



Universidad de las Ciencias Informáticas

Facultad 6

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

***Título: Componente web de visualización en 3D
para objetos geo-referenciados en el SIG UCI***

Autor:

Gabriel Padrón Castillo

Tutor:

Ing. Yadir Hernández Batista

La Habana, mayo de 2014“

Año 56 de la Revolución”

“Todas las ideas esenciales de la Ciencia han nacido de un conflicto dramático entre la realidad y nuestro deseo de comprenderla.”

A. Einstein

*Componente de visualización de objetos en 3D sobre la web para
SIG UCI*

Declaración de autoría

Se declara por este medio que, Gabriel Padrón Castillo con carnet de identidad 90101737386, es el autor de este trabajo y autoriza a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firma la presente declaración jurada de autoría en La Habana a los ____ días del mes ____ del año _____

Yadir Hernández Batista

Firma del Tutor

Gabriel Padrón Castillo

Firma del Autor

Tutor: Yadir Hernández Batista.

Correo Electrónico: yadirhb@uci.cu

Especialidad: Ingeniero en Ciencias Informáticas.

Año de Graduación: 2013.

Institución: Departamento GEYSED, UCI.

Desempeño: Diseñador de interfaz de usuario en el proyecto Aplicativo SIG.

Dedicatoria

A mi abuelo que desde mi nacimiento estuvo a mi lado, atento a todo alrededor para guiarme, apoyarme, y que desafortunadamente no pudo estar en el transcurso de toda mi carrera universitaria, su influencia hubiese sido de gran importancia para mi formación como profesional.

A mi madre por inculcarme la inquietud por el conocimiento, la aspiración de superarme y el valor de luchar por ser un profesional que sepa dar solución a los diferentes problemas del contexto.

A mi padre por seguir cada día mis pasos y orientarme para obtener los resultados que se esperan de un joven de estos tiempos.

A mi abuela por apoyarme desde niño en todas las tareas que de una forma u otra tienen que ver con la obra que defiendo.

A Ernesto por contribuir a mi formación, siendo un padre para mí.

A mis hermanos Julio Cesar, Javier y Daniel.

A mis tías Gisel, Gloria, Tania.

A mi familia de la Habana: Ana que siempre estuvo atenta a todos los aspectos vinculados a mi alimentación, Mayra, Pipo, Mirian, Dory que me dieron el apoyo durante todos los años de la Universidad.

A todo el que me ayudó de alguna u otra forma.

Agradecimientos

A todos los docentes de la Universidad de Ciencias Informáticas que contribuyeron a mi formación durante los cinco años.

A mi hermano Julio Cesar por su colaboración en la búsqueda de información.

Al profesor Agustín López Perdigón por sus sabias reflexiones.

Al compañero “El Tropa” de UCIFAR por sus horas dedicadas a orientarme en la realización de esta investigación.

A mis compañeros de estudio Raydel, Lester, Juan Carlos, David, Otero, Fidelky, Julio, Celia, Arlettys, Mavis, Martica, Sandy, por ofrecerme el apoyo para culminar lo que nunca parecía tener un final.

A los miembros del proyecto Aplicativo SIG del cual formo parte, Listley, Alain, Yadir, Yenier, por brindar su apoyo.

De forma general a todos lo que me ayudaron.

Resumen

A partir de comprender la importancia de los Sistemas de Información Geográfica (SIG) y sus diversas aplicaciones en aspectos de orientación, consulta, gestión de procesos socio-económicos y administrativos a varios niveles, diversos investigadores han profundizado en el tema y han elaborado numerosos SIG. Particularmente la Universidad de las Ciencias Informáticas en Cuba cuenta con uno de estos sistemas: el SIG UCI, el cual ofrece numerosas ventajas a los usuarios de la Universidad dado que les permite identificar la ubicación de distintos sitios de interés en la institución, la cual cuenta con un amplio número de edificaciones, además posibilita la ejecución de cálculos de distancia, de áreas y la localización de personas y edificios. Independientemente de las potencialidades del SIG UCI, se evidencia la existencia de aspectos que limitan las capacidades del sistema para la visualización, queda una brecha de información entre la necesidad del usuario y la información que ofrece el sistema al solo mostrar la localización de edificios en el mapa, impidiendo que los usuarios aprecien propiedades de las edificaciones como forma, número de pisos, es por esto que se decide desarrollar la presente investigación. Una adecuada representación de las edificaciones geo-referenciadas en el SIG UCI es fundamental para solventar las fisuras en la información, de ahí la importancia que tiene esta investigación dirigida a desarrollar un componente que permita mejorar la representación de las edificaciones geo-referenciadas en el SIG UCI a través de la visualización de estructuras geográficas en 3 dimensiones (3D).

Palabras clave:

Sistema de Información Geográfica, edificaciones geo-referenciadas, visualización en 3D.

Abstract

Based upon the comprehension of the importance of the Geographical Information Systems (GIS) and its various applications in the orientation, consultation, and negotiation of the administrative and socio-economic processes at different levels, several researchers have gone deeper into this topic, and they have elaborated the GIS. The University of Informatics Sciences in Cuba has one of these systems, termed the SIG UCI, which shows some advantages to the university users, for it allows identifying the location of various sites of interest in the institution, which has many buildings. It makes possible distance and area calculations, and also the location of people and buildings. Apart from the SIG UCI potentialities, there are some elements that hinder the capacity of the visualization system; there is an information gap between the necessity of the user and the information given by the system while showing the location of the buildings with their corresponding form and number of the floors, being the main goal of this paper. A reasonable representation of the geographically-referenced buildings in the SIG UCI is essential to solve the scarcity of information; thus the importance of this research works: it is targeted to develop a component to improve the representation of the buildings in the SIG UCI throughout the visualization of geographic structures in three dimensions (3D).

Keywords: Geographical Information Systems, geographically-referenced buildings, 3D visualization.

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Índice

Índice

Introducción	1
Capítulo I: Fundamentación teórica de la representación de información geográfica en los SIG en 3D.....	7
1.1 Introducción.	7
1.2 Conceptos asociados al dominio del problema.....	7
1.3 Objeto de estudio.....	9
1.3.1 Descripción General.....	9
1.3.2 Estándares OGC para la gestión de información geográfica.....	11
1.3.3 Vinculación de los SIG con las herramientas de diseño.....	15
1.3.4 Análisis de soluciones existentes que permitan la modelación en 3D a partir de datos geográficos.....	15
1.4 Conclusiones parciales.....	19
Capítulo II: Tendencias y tecnologías actuales para el desarrollo del componente de visualización.....	20
2.1 Metodologías de desarrollo de software.....	20
2.1.1 El Proceso Unificado de Desarrollo de Software (RUP).....	20
2.2 Lenguaje Unificado de Modelado (UML).....	22
2.3 Herramientas Case.....	22
2.3.1 Herramientas Case Visual Paradigm for UML Enterprise Edition 8.0.....	22
2.4 Lenguajes de programación.....	23
2.4.1 Preprocesador de Hipertexto (PHP) 5.....	23
2.4.2 JavaScript.....	23
2.5 Bibliotecas de clases para la representación tridimensional.....	24
2.5.1 WebGL.....	24

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Índice

2.5.2 Bibliotecas de clases Three.js.....	24
2.6 Framework de Desarrollo.....	25
2.6.1 Framework de Java Script Ext JS 4.2.1.....	25
2.7 Sistemas Gestores de Bases de Datos (SGBD) PostgreSQL 9.1.....	26
2.7.1 PostGIS 1.5.8.....	26
2.7.2 Sistema Administración de Bases de Datos pgAdmin III 1.14.0.....	26
2.8 Entorno de Desarrollo Integrado (IDE).....	27
2.8.1 NetBeans 7.4.....	27
2.9 Servidor Web Apache 2.2.....	27
2.10 Conclusiones parciales.....	27
Capítulo III: Descripción de la solución propuesta.....	29
3.1 Introducción:.....	29
3.2 Modelo del dominio.....	29
3.2.1 Descripción del diagrama del modelo dominio.....	31
3.3 Requisitos de software.....	31
3.3.1 Técnica de extracción de requisitos.....	31
3.3.2 Requisitos funcionales.....	32
3.3.3 Requisitos no funcionales.....	32
3.4 Descripción del sistema propuesto.....	34
3.4.1 Descripción de los actores del sistema.....	34
3.4.2 Diagrama de Casos de Uso del Sistema.....	35
3.4.3 Descripción textual de los Casos de Uso del Sistema.....	35

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Índice

3.5 Conclusiones parciales.....	38
Capítulo IV: Construcción de la solución propuesta.....	39
4.1 Introducción.	39
4.2 Arquitectura del software.	39
4.2.1 Estilos Arquitectónicos.....	39
4.2.2 Patrones Arquitectónicos.....	40
4.3 Patrones de Diseño.....	41
4.3.1 Patrones de Diseño GRASP.	42
4.3.2 Patrones de diseño GoF.	44
4.4 Modelo de diseño.....	44
4.4.1 Diagrama de Clases del Diseño (DCD).	45
4.5 Diseño de la Base de Datos	46
4.5.1 Modelo entidad-relación.	46
4.6 Modelo de Implementación.	47
4.6.1 Diagrama de Componentes.	48
4.7 Modelo de Despliegue.....	48
4.8 Pruebas de software.....	49
4.8.1 Pruebas de Caja Negra.	50
4.8.2 Resultados de la aplicación de las Pruebas de Caja Negra.....	52
4.9 Conclusiones Parciales.....	52
Conclusiones Generales.....	54
Recomendaciones.	55

*Componente de visualización de objetos en 3D sobre la web para
SIG UCI*

Índice

Referencias	56
Bibliografía.....	59

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Índice de Figuras

Fig. 1: Diagrama del modelo de dominio.	30
Fig. 2: Diagrama de Casos de Uso del Sistema.	35
Fig. 3: Diagrama de clases del diseño.	45
Fig. 4: Modelo entidad relación parte 1.	46
Fig. 5: Modelo entidad relación parte 2.	47
Fig. 6: Diagrama de componentes.	48
Fig. 7: Diagrama de despliegue.	49
Fig. 8: Resultado de las pruebas de caja negra.	52

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Índice de Tablas

Tabla 1: Descripción de los actores del sistema	34
Tabla 2: Descripción del caso de uso visualizar edificio 3D.	35
Tabla 3: Secciones a probar en el caso de uso.....	50
Tabla 4: Descripción de las variables de entrada.....	51
Tabla 5: Matriz de datos SC1	51
Tabla 6: Matriz de Datos SC2.....	52

Introducción

Durante el proceso evolutivo el hombre ha tenido que localizar elementos en el territorio que habita para satisfacer sus necesidades; el aumento de conocimientos sobre el uso de información geográfica para ubicarse en el terreno ha conducido hacia la búsqueda de distintas formas de almacenarla, comprenderla, representarla y posibilitar su utilización.

Estudios realizados demuestran el continuo perfeccionamiento de las formas de representación de la información geográfica. Una de las primeras formas empleadas por el hombre se basó en la realización de pinturas de los animales que cazaba en paredes de cuevas, diversas investigaciones coinciden que estas tenían como objetivo referenciar las rutas migratorias de los animales, lo cual facilitaba las labores de caza. A medida que fue evolucionando ideó una nueva forma de proyectar la información geográfica: el mapa, una representación plana de la tierra con una escala que constituye el nivel de proporcionalidad que tiene la imagen del territorio plasmada en él con respecto al territorio real y permite referenciar todo tipo de lugar. La creciente necesidad de elaboración de mapas condicionó el surgimiento de la cartografía como ciencia que se encarga del trazado y el estudio de mapas geográficos.

Los mapas constituyen una herramienta importante para la humanidad, en la actualidad son más detallados, exactos y brindan variedad de información. Pueden encontrarse en formato duro y digital, este último brinda un sinnúmero de ventajas pues facilita el acceso a la información y diversifica las opciones para la interacción con los mismos.

El desarrollo actual de las TIC¹ ha revolucionado la forma de almacenar y utilizar la información geográfica, una muestra de esto son los Sistemas de Información Geográfica (SIG), los cuales pueden definirse como una unidad integrada de hardware, software y datos geográficos que permite capturar, almacenar, manipular, analizar y desplegar la información geográficamente referenciada para su utilización en diversas esferas de la realidad.

¹ TIC Tecnologías de la Información y las Comunicaciones.

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Introducción

El término SIG puede definirse además como: *“un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones”* (HARVEY, 2010).

Estos sistemas poseen numerosas potencialidades que son aplicadas en aspectos de orientación, consulta y gestión de procesos socio-económicos y administrativos a varios niveles, de ahí que diversas instituciones en el mundo han profundizado en el estudio, creación y despliegue de aplicaciones de este tipo.

En Cuba, la Universidad de las Ciencias Informáticas ha desarrollado varios SIG, entre los que se encuentra el SIG UCI, esta herramienta fue creada por las necesidades de localización de lugares dentro de la institución. Las características propias de este centro justifican la creación de este SIG, al poseer una amplia extensión superficial con más de 100 edificaciones, contar con una matrícula superior a los 5000 estudiantes de pregrado y una cifra considerable de trabajadores, distribuidos en diversas áreas: docentes, productivas, residenciales y otras tales como: complejo comedores, hospital, edificio de televisión y radio universitaria.

El SIG UCI ofrece múltiples beneficios a los usuarios internos y al diverso número de personas externas que acceden para participar en actividades de todo tipo dentro de la institución. Posee información sobre la localización de lugares de interés dentro de la Universidad, lo que propicia orientar a los usuarios a la hora de dirigirse hacia alguno de estos. Cuenta con numerosas funcionalidades que permiten realizar cálculos de distancia, elemento de importancia para la planificación del tiempo de traslado de un lugar a otro, así como el cálculo del área que abarca determinado terreno, el cual puede ser empleado para la toma de decisiones en aspectos vinculados a la realización de obras de construcción, además posee información asociada al lugar de residencia de estudiantes y trabajadores internos, esta funcionalidad constituye una forma de localizar a determinada persona y de identificar el camino que debe recorrerse para contactar con esta.

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Introducción

A pesar de las disímiles ventajas que proporciona, existen varios aspectos que limitan las capacidades del sistema, por ejemplo: al representar la información de los edificios en dos dimensiones (2D), el usuario no es capaz de apreciar la forma y entorno real de los mismos, no se logra identificar desde el punto de vista visual la altura y cantidad de pisos de los edificios lo que priva a los usuarios de conocer la localización de un área dentro de los edificios.

Los aspectos referidos con anterioridad marcan un límite de detalle para la herramienta mencionada que llega solo a nivel de las edificaciones. Esto conlleva a que se conviertan zonas de importancia en zonas oscuras, pues queda una brecha de información entre la necesidad de los usuarios y la información brindada por el sistema al solo mostrar la localización de un edificio en el mapa de la Universidad, lo cual no permite saber la forma que tiene, ni orientarse con exactitud hacia determinado piso, de esta forma se limita su localización detallada y acceso con mayor rapidez, esta situación converge en una problemática para la comunidad UCI.

Una vez descrita la situación actual en que se encuentra el SIG UCI y haber identificado deficiencias se plantea el siguiente **problema a resolver**: ¿Cómo mejorar la representación de las edificaciones geo-referenciadas en el SIG UCI?

Para dar solución al problema formulado se plantea como **objetivo general**: Desarrollar un componente que permita mejorar la representación de las edificaciones geo-referenciadas en el SIG UCI a través de la visualización de estructuras geográficas en 3 dimensiones (3D).

Se determina como **objeto de estudio**: El proceso de representación de las edificaciones geo-referenciadas en los SIG y como **campo de acción**: la representación de las edificaciones en 3D en el SIG UCI.

Se plantea como **idea a defender**: Si se desarrolla un componente que permita representar las edificaciones en 3D en el SIG UCI aumentaría el nivel de detalles para la localización de lugares de interés dentro de la Universidad por parte de los usuarios.

Para cumplir los objetivos antes expuestos se trazan las siguientes **tareas de investigación**:

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Introducción

1. Determinación de los fundamentos sobre formas de representación de la información geográfica, con el objetivo de identificar sus características y posibilitar la comprensión de su funcionamiento.
2. Análisis de las formas de representar objetos en tres dimensiones en los SIG, se realiza con el fin de apreciar las características de este tipo de sistemas para analizar si son aplicables al problema identificado.
3. Valoración de las herramientas, metodologías y lenguajes a utilizar en la construcción de la solución, esta tarea se plantea con el objetivo de realizar una correcta elección de las tecnologías que facilitarían el desarrollo del componente.
4. Implementación del componente de visualización de edificaciones geo-referenciadas en 3D.
5. Modelación de los distintos artefactos que genera la metodología de desarrollo de software seleccionada, dicha tarea se traza con el interés de plasmar los aspectos fundamentales del proceso de construcción de la solución, a partir de la confección de la documentación referente a estos procesos.
6. Realizar las pruebas a la herramienta desarrollada, tarea que se encuentra destinada a comprobar el correcto funcionamiento de la herramienta.

En el presente trabajo se utilizaron diversos métodos de investigación teóricos y empíricos, con el objetivo de identificar vías de solución al problema planteado a partir de la comprensión de sus características, estos se describen a continuación:

Métodos teóricos:

Análítico-Sintético: Permitió realizar un análisis de la bibliografía referente a los SIG en sitios web, libros, tesis, con la intención de obtener información de los elementos básicos del objeto y el campo de investigación, permitió además realizar una síntesis detallada de la misma donde quedan recogidos los aspectos principales que conforman la visualización de elementos en los SIG, basados en entornos web. También posibilitó la elaboración de las conclusiones parciales y finales.

Análisis Histórico-Lógico: Se utilizó para conocer la trayectoria de los SIG, la existencia de estos en 2D y 3D, sus características, así como los requisitos de hardware necesarios para su utilización.

Componente de visualización de objetos en 3D sobre la web para SIG UCI

Introducción

La Inducción-Deducción: Permitió realizar generalizaciones sobre la creación de gráficos en 3D en entornos web a partir de datos geográficos, a partir del estudio de herramientas existentes.

Modelación: Posibilitó modelar los distintos diagramas que explican el proceso de construcción del componente, para representar las edificaciones en 3D.

Métodos empíricos:

Observación: Este método se utilizó para constatar las funcionalidades y procedimientos de las herramientas ya implementadas en torno al objeto de estudio.

Entrevista: Se utilizó con el objetivo de conocer las principales experiencias de especialistas en la representación de objetos 3D en la web y en particular de la información geográfica.

Para la aplicación de este método se seleccionó como población a los profesores del proyecto Aplicativos-SIG, así como especialistas del centro XETIC, esto permitió obtener la información necesaria para dar solución al problema planteado.

Se utilizó la técnica de muestreo no probabilístico para seleccionar la muestra, específicamente el muestreo intencional el cual permite seleccionar los integrantes de la muestra, sobre la base de la elección de las personas que puedan aportar mayor cantidad de información a partir de su experiencia.

Estructura de la investigación:

La presente investigación científica está compuesta por introducción, cuatro capítulos, conclusiones, recomendaciones, bibliografía y anexos. A continuación se describen cada uno de ellos:

En el capítulo 1 “**Fundamentación teórica de la representación de la información geográfica en los SIG en 3D**” se hace referencia a los conceptos asociados al dominio del problema y se realiza un análisis de las soluciones existentes. Durante el desarrollo del capítulo 2 “**Tendencias y tecnologías actuales para el desarrollo del componente de visualización**” se analizan las herramientas, tecnologías y metodología de desarrollo de software, necesarias para el correcto desarrollo del sistema informático que se desea implementar. El capítulo 3 “**Descripción de la solución propuesta**” versa sobre la descripción

Introducción

de la solución a través del diagrama del modelo del dominio, el levantamiento de requisitos funcionales y no funcionales, así como diagrama de casos de uso. El capítulo 4 “**Construcción de la Solución Propuesta**” aborda todo el proceso de construcción de la solución escogida para la solución del problema planteado.

Capítulo I: Fundamentación teórica de la representación de información geográfica en los SIG en 3D.

1.1 Introducción.

En el presente capítulo se realiza un análisis de las principales características y conceptos que se encuentran vinculados a los SIG y a la visualización de objetos en 3D sobre la web, a partir de reconocer sus ventajas en función de lograr una mejor localización de las edificaciones en la UCI. Además se fundamenta el estado en que se encuentran los sistemas que muestran información geográfica en 3D en el mundo y particularmente en Cuba.

1.2 Conceptos asociados al dominio del problema.

Diversos autores han enfatizado en el uso del mapa en distintos contextos: *“es reconocido como un objeto de vital importancia para la obtención de información, son insustituibles para el establecimiento de relaciones espaciales y dependencia de los fenómenos, ellos constituyen la más idónea ilustración de la distribución espacial de objetos, fenómenos o procesos naturales, socioeconómicos, históricos o de otro tipo, en la superficie terrestre”* (CRUZ, 2009).

Los SIG se basan en el principio de utilización de los mapas, constan de datos geográficos almacenados en bases de datos, los cuales son empleados para representar y relacionar cualquier tipo de información geográfica y garantizar la creación de consultas interactivas para el análisis de dicha información, estas razones justifican la necesidad de profundizar en la conceptualización que se tiene del mapa.

Mapa

“Los mapas geográficos son representaciones planas de la superficie terrestre en dos dimensiones: largo y ancho. El mapa geográfico representa toda la superficie terrestre (Planisferio) así como una parte de ella”(GÁRATE et al., 2009).

“Un mapa es una representación geográfica de una parte de la Tierra o de su totalidad en una superficie plana. Tradicionalmente, los mapas son simplemente representaciones estáticas y en dos dimensiones de un espacio en tercera dimensión, mientras que otros, especialmente los que han surgido con la popularización de las nuevas tecnologías, son dinámicos, interactivos e incluyen representaciones en tercera dimensión de lugares de interés” (ALIAGA, 2006).

El análisis de estas definiciones permite identificar puntos coincidentes entre los autores en cuanto al tipo de representación geográfica que realizan, sin embargo, la última incluye aspectos vinculados al uso de las tecnologías con nuevas potencialidades para la obtención de mayor volumen de información que ofrece ventajas a los usuarios.

La elaboración de mapas se sustenta sobre la base de la cartografía, significa que el dominio teórico y práctico de esta es imprescindible para obtener y referenciar información de la ubicación de determinado lugar de interés en el mapa, a partir de coordenadas geográficas.

Cartografía

Según se plantea en (ALARCÓN, 2010) “La cartografía es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales. Además se puede definir como el conjunto de mapas producidos por una institución, relativos a un determinado territorio. Es la técnica geográfica que estudia la secuencia de etapas y procesos ejecutados para la visualización de un espacio geográfico mediante la producción de mapas, cartas, planos o croquis”.

La cartografía es una ciencia “cuyo fin es ofrecer una representación gráfica fiable del territorio; mostrar de forma reducida, sintética y esquemática una realidad compleja e incluso a veces difícilmente abarcable, y a través de ello obtener una mejor comprensión de los fenómenos de la Tierra, su localización, tamaño y distribución” (ROBINSON et al., 1987).

Ambas definiciones permiten comprender el significado de la cartografía como ciencia fundamental para la proyección de mapas que posibiliten representar gráficamente un territorio determinado con el fin de conocer a profundidad las características que lo tipifican para su posterior uso en las diferentes esferas de la realidad. A través de la cartografía se puede obtener la información necesaria en torno a la ubicación

geográfica de determinada edificación, dicha información puede ser utilizada para la elaboración de modelos en 3D y de esta forma proporcionarle a los usuarios una mejor localización de estas edificaciones.

Visualización en 3D

“Consiste en todas las operaciones que conlleva el despliegue de un objeto tridimensional en un dispositivo gráfico de salida. Por lo general, el objeto es concebido como una serie de triángulos, que serán procesados por un motor gráfico 3D, con soporte de hardware gráfico” (COMA et al., 1998).

Diversos autores expresan sus opiniones acerca de la visualización en 3D, aunque no se basan en una definición específica, coinciden en que se sustenta en la proyección de objetos tridimensionales sobre un dispositivo de salida. Su utilización ha cobrado una importancia significativa, al mejorar considerablemente la visualización de estos objetos y permitir apreciar características que no son distinguibles en una proyección bidimensional.

1.3 Objeto de estudio.

En este epígrafe se presentan los aspectos fundamentales vinculados con el objeto de estudio de esta investigación, orientado al proceso de representación de las edificaciones geo-referenciadas en el SIG UCI. En un primer momento se efectúa una descripción del mismo desde el punto de vista general y posteriormente se analizan las soluciones existentes.

1.3.1 Descripción General.

Los SIG han revolucionado la forma de representar la información geográfica, son numerosas las facilidades que brindan: rapidez, accesibilidad, mejor nivel de detalles, fácil almacenamiento y actualización de los datos, permiten el análisis de patrones, relaciones y tendencias en la información, todo con el interés de contribuir a la toma de mejores decisiones. El uso de SIG facilita la visualización de los datos obtenidos en un mapa con el fin de reflejar y relacionar fenómenos geográficos de cualquier tipo. Estos agrupan en distintas capas la información geográfica, a medida que se realiza el acoplamiento de

estas se muestra el panorama visual resultante, que permite el uso del sistema como referencia geográfica.

La creación de SIG basados en entornos web ha aumentado considerablemente, esto posibilita que un mayor número de personas tenga acceso a las funcionalidades que brindan, sin necesidad de tener toda la información geográfica almacenada y de conocer el funcionamiento interno.

Los SIG han utilizado dos modelos fundamentales para representar información geográfica, los cuales permiten el almacenamiento de los datos: el modelo vectorial y el modelo *raster*. El primero es una estructura de datos utilizada para almacenar datos geográficos. Los datos vectoriales constan de líneas o arcos definidos por sus puntos de inicio y fin así como puntos donde se cruzan varios arcos: los nodos. Un SIG vectorial se define por la representación vectorial de sus datos geográficos. De acuerdo a las peculiaridades de este modelo de datos, los objetos geográficos se representan explícitamente y, junto a sus características espaciales, se asocian sus valores temáticos. En dicho modelo el interés de las representaciones se centra en la precisión de localización de los elementos geográficos sobre el espacio (GARCIA, 2011).

El segundo modelo constituye una codificación de los datos geográficos, en la que se representa el valor medio o más representativo de una porción regular del territorio en una porción equivalente y escalada en el mapa digital; las porciones del territorio y su representación digital suelen constituir una malla regular de cuadriláteros (NAVARRO, 2011).

Los dos modelos citados tienen ventajas en la representación de la información geográfica, el vectorial posee las siguientes: una estructura de datos muy compacta, codifica eficientemente la tipología, la salida en papel presenta muy buenos productos. Por su parte el modelo *raster* ofrece numerosas ventajas entre las que se pueden destacar: utiliza una estructura de datos muy simple, las superposiciones de las diferentes coberturas se implementan de forma rápida y eficiente, permite una forma más eficiente de representación cuando la variación espacial es muy alta, es apropiado para el tratamiento de imágenes de satélite, brinda la posibilidad de generar modelos de elevación del terreno.

El SIG UCI forma parte de los productos desarrollados por el proyecto aplicativo SIG del Centro de Geoinformática y Señales Digitales (GEYSED). Este sistema ofrece una panorámica en dos dimensiones de la

Universidad mediante la proyección del mapa geográfico del área que esta ocupa, brinda servicios que permiten obtener una referencia al localizar determinado lugar dentro del territorio que abarca la institución mencionada, en dependencia de los datos que se tenga de los mismos.

Las potencialidades del SIG UCI se ven limitadas al proyectar el mapa en 2D, lo que impide a los usuarios la apreciación de la forma real, altura, y cantidad de pisos de los edificios, en cambio una proyección tridimensional de las edificaciones ubicadas en esta institución, ofrecería un nivel más detallado y una mejoría de la geo-referenciación de los sitios de interés. Esto implica que se profundice en la búsqueda de alternativas que lo posibiliten a partir de las soluciones ya elaboradas que sirvan como punto de apoyo.

Estudios recientes han estado destinados a la vinculación de la visualización en 3D con información geográfica para contribuir a mejorar la localización de sitios de interés. A partir de la materialización en la práctica de la idea referida se han desarrollado SIG que añaden a su funcionamiento la proyección de objetos en 3D; esto hace posible obtener nuevas perspectivas, al brindar apreciaciones que no son visibles en un mapa plano, como es el caso de edificios, torres, elevaciones del terreno, es decir, la visualización en 3D permite un enfoque real de los objetos en el espacio.

1.3.2 Estándares OGC para la gestión de información geográfica.

Open Geoespacial Consortium (OGC) es un consorcio internacional que tiene como propósito la implementación de estándares de interfaces para promover y facilitar el uso global y mundial de la información espacial, los cuales permiten a los desarrolladores crear sistemas de información que pueden fácilmente intercambiar información geográfica e instrucciones con otros sistemas de información. Los estándares creados por esta institución se agrupan en estándares para: visualizar datos, acceder a datos, codificar datos, catálogos y registros, y el procesado de datos (CONSORTIUM, 2013).

Algunos de estos estándares pueden ser utilizados para la visualización de información geográfica en 3D. A continuación se realiza una breve descripción de varios de los estándares de la OGC de forma general, y específicamente los vinculados a la representación tridimensional, con el objetivo de conocer las posibilidades de aplicación de estos al problema identificado:

WMS (*Web Map Service*): Define los elementos necesarios para un servicio de mapas. Un servicio WMS devuelve una imagen con información geográfica, pero esta solo contiene la propia información visual para que el cliente pueda mostrarla (OLAYA, 2011).

WFS (*Web Feature Service*): Se encuentra relacionado con los datos de tipo vectorial, y a través de él se sirven directamente las entidades de un dato vectorial con sus geometrías y datos alfanuméricos asociados. Desde este punto de vista, acceder a un servicio WFS es similar a acceder a una capa vectorial cualquiera o a una base de datos, ya que el SIG puede recuperar la información correspondiente (tanto la componente geográfica como la temática de cada entidad) y operar con ella (OLAYA, 2011).

WCS (*Web Coverage Service*): Permite obtener los datos *raster* de un servidor en forma de entidades. Este servicio está diseñado para tratar representaciones de un fenómeno que varía en el espacio (OLAYA, 2011).

GML (*Geography Markup Language*): es un lenguaje de marcado basado en XML para el modelaje, transporte y almacenamiento de información geográfica. Utilizando este lenguaje, resulta posible el intercambio de información geográfica de forma interoperable. Es un lenguaje extremadamente genérico, que permite recoger tanto datos *raster* como vectoriales y hacerlo con mucha flexibilidad (OLAYA, 2011).

KML (*Keyhole Markup Language*): es un lenguaje de marcado basado en XML para representar datos geográficos en 3D. Inicialmente fue utilizado por Google Earth para la geo-referenciación de distintos recursos en su visor, dado que este lenguaje soporta una serie de primitivas geométricas como son el punto, líneas, polígonos o incluso modelos collada. Por si solo no constituye un visor de información geográfica tridimensional, sino que propicia una vía de organización y acceso a distintos recursos geográficamente referenciados que posteriormente puede ser utilizada por distintas herramientas de visualización (REBOLLO and FERNÁNDEZ, 2011).

CityGML: Constituye un modelo de datos estándar definido para la representación de modelos de ciudad 3D que combina información semántica y geométrica. El objetivo del desarrollo de City GML es llegar a una definición común de las entidades básicas, atributos y relaciones de un modelo de ciudad 3D, se trata de una aplicación del *Geography Markup Language* (GML3) que permite el intercambio de datos espaciales. Propone la aplicación de una metodología para crear ciudades 3D a partir de la aplicación de

servicios Web que implementen interfaces estándar del OGC para el acceso a datos tales como: WMS, WFS, WCS. Este estándar se caracteriza por contemplar cinco niveles de detalle con lo cual es posible crear modelos de ciudad 3D en que el grado de apreciación visual varía, en dependencia de los datos con que se cuente (KOLBE, 2009).

La aplicación de este estándar representa un avance significativo en cuanto a organización estructural para la construcción de modelos de ciudades 3D, sin embargo uno de los inconvenientes de la generación de estos modelos es que requiere una inversión de horas y costos muy altos. Existen múltiples tecnologías de adquisición de información para la creación de ciudades 3D como el escaneado laser, fotogrametría terrestre y la aplicación de herramientas de modelado, las cuales deben ser empleadas para la aplicación de este estándar, aspectos que inducen a la afirmación del planteamiento anterior.

WVS (Web View Service): Constituye un servicio de interpretación de datos geográficos en tres dimensiones, tales como modelos de paisaje, modelos de ciudad, modelos de vegetación, o los modelos de infraestructura de transporte. El WVS encapsula todo el proceso de interpretación del lado del servidor y proporciona múltiples capas de imágenes renderizadas de escenas 3D para el consumidor del servicio.

Este servicio es una extensión del Servicio Web de Terreno (WTS) y del Servicio Web de vista Perspectiva (WPVS), cuyos principios básicos están destinados a producir vistas en perspectiva de datos geo-referenciados, estandarizando la forma en la cual el cliente solicita la vista y en la que el servidor procesa la petición y retorna la información correspondiente, de esta forma se puede realizar la proyección desde distintos ángulos de la vista renderizada en formatos de imagen como PNG, GIF, JPEG. El grado de usabilidad de estos dos servicios se ve limitado debido a que no poseen capacidades de interacción o navegación. WVS contrarresta las desventajas de los servicios mencionados con anterioridad, dado que proporciona datos geométricos y temáticos adicionales, la información de identidad de objeto, que se codifica en las imágenes de múltiples capas, además proporciona operaciones para la recuperación de información sobre los objetos visualizados en posiciones específicas de imagen, funcionalidades de medición, y soporte mejorado de navegación (NEUTENS and DE MAEYER, 2010).

Es importante destacar las continuas investigaciones que realizan las distintas organizaciones, institutos y universidades para crear y perfeccionar formas de incorporar la visualización tridimensional a los SIG

producto a las ventajas que ofrece y a la gran aceptación que presenta por parte de los usuarios. Actualmente existen nuevos servicios que han quedado registrados en especificaciones de la OGC para su futura estandarización, entre ellos se encuentran:

W3DS (3D Web Service): Representa un servicio de interpretación de datos geográficos en 3D, tales como modelos de paisaje, los modelos de la ciudad, la construcción de modelos aplicando texturas, la vegetación y el mobiliario urbano. Los datos se agrupan en escenas que se componen de elementos de visualización, se garantiza el renderizado eficiente en tiempo real a altas velocidades de fotogramas. El W3DS es capaz de manejar conjuntos de datos de una amplia gama de escalas y estos pueden constar de varios niveles de detalle para cada objeto, lo que aumenta en gran medida el rendimiento sin sacrificar la calidad (KLIMKE and DÖLLNER, 2011).

SLD 3D (Style Layer Descriptor 3D): El SLD 3D no es un estándar aprobado por la OGC, aunque sí es una propuesta recogida por este organismo para extender el estándar actual SLD de manera que englobe la representación de escenas en tres dimensiones. Gracias a la definición del SLD 3D se ha logrado la inclusión de la tercera componente espacial a partir de la información existente en los elementos 2D, permitiendo la generación automática de volúmenes, mediante la extrusión de los datos 2D. De esta forma se logra la generación aproximada de las edificaciones a partir de la información en dos dimensiones, la generación de las infraestructuras existentes o la visualización de mapas temáticos en 3D entre otras aplicaciones y usos (REBOLLO and FERNÁNDEZ, 2011).

Una vez descritos algunos de los principales estándares recogidos por OGC para la visualización de información geográfica, se pueden identificar las ventajas que presentan a partir de que proveen una estructura para lograr la visualización tridimensional. Sin embargo cabe destacar que la implementación de varios de estos estándares como servicios web, requiere que el servidor presente una gran capacidad de procesamiento para atender las peticiones, en caso de no cumplir esta condición, se realizaría una sobrecarga de este, pudiendo colapsar en dependencia de la cantidad de solicitudes realizadas. Otros aspectos que influyen en la implementación de estos servicios son los datos con que se cuenta, el tiempo y recursos necesarios, los cuales determinan si es factible aplicar dichos estándares.

1.3.3 Vinculación de los SIG con las herramientas de diseño.

Existen diferentes formas de representar objetos 3D en entornos web, se puede destacar la utilización de *plugin* para flash en los navegadores, la importación de objetos hechos en programas especializados en el diseño como Blender, 3DMax, que posibilitan la creación de objetos con un nivel de detalle relativamente alto. Con el interés de mejorar aspectos relacionados a la visualización de entornos urbanos, grandes compañías como Google han enfatizado en establecer vínculos entre los programas de diseño y los SIG. En el caso específico de Google, brinda la herramienta de diseño Google SketchUp, con la cual es posible crear modelos de ciudad 3D, a partir del diseño de estructuras tridimensionales como edificios y torres.

Una vez creados los modelos 3D, los programas de diseño permiten exportarlos a formatos como: obj, collada que pueden ser importados en entornos web, los cuales se les aplica un proceso de escalabilidad, y son incluidos en SIG basados en modelos *raster*. El panorama visual resultante presenta características similares al ambiente real. El empleo de este método implica modelar los objetos individualmente y requiere de un tiempo considerable.

1.3.4 Análisis de soluciones existentes que permitan la modelación en 3D a partir de datos geográficos.

A continuación se describen algunos sistemas que permiten la visualización tridimensional de información geográfica, con la intención de identificar la factibilidad de su empleo. La profundización del estudio de las características de estas soluciones constituye una tarea fundamental para establecer puntos de referencia, que permitan guiar la construcción de una alternativa de modelación de edificaciones 3D para el SIG UCI, en caso de que estas no sean aplicables.

Google Earth

Google Earth es una herramienta que garantiza ver, crear y compartir información geográfica, que posteriormente se puede explorar en una interfaz interactiva y visualmente intuitiva. Proporciona una gran riqueza de imágenes e información geográfica y astronómica. Presenta una capa de edificios en 3D con alto nivel de detalle. Sus funcionalidades permiten:

- Observar imágenes 3D de alta resolución de ciudades, montañas y valles, complementados con información relevante.
- Observar miles de puntos de datos creados por otros usuarios.
- Representar datos de ubicación mediante herramientas de dibujo 3D o transferir hasta 2.500 ubicaciones por direcciones o coordenadas geoespaciales desde una hoja de cálculo.
- Incorporar datos en formatos de archivo como .shp y .tab a través del módulo de importación de datos. Algunos ejemplos son los datos de parcelas, demográficos y de edificios 3D (PATTERSON, 2007).

Cesium

Cesium es una biblioteca de JavaScript para la creación de globos en 3D y mapas 2D en un navegador, sin necesidad de un *plugin*. Esta herramienta se caracteriza por: el empleo de WebGL², tecnología que garantiza la eliminación de dependencias adicionales para llevar a cabo el proceso de visualización, ser multiplataforma y permitir la visualización de datos dinámicos. Cesium constituye una herramienta de código abierto bajo la licencia Apache 2.0, gratuito para el uso comercial y no comercial y compatible con un globo 3D, mapas en 2D y 2.5D. Esta librería posibilita:

- La visualización dinámica de datos geoespaciales.
- Visualizar terrenos de cualquier zona geográfica a partir de múltiples fuentes.
- Dibujar capas de imágenes utilizando WMS, TMS, OpenStreetMaps, Bing, y las normas de ESRI.
- Dibujar los datos vectoriales a partir de KML y ESRI Shapefiles.
- Dibujar poli líneas, polígonos, círculos, iconos, etiquetas y objetos personalizados.
- Utilizar el sistema de materiales para cambiar su apariencia.
- Control de cámara y crear rutas de vuelo.
- Utilizar rutinas geométricas y de representación de bajo nivel (SELLERS *et al.*, 2013).

OpenWebGlobe

² WebGL Librería Gráfica en la Web

OpenWebGlobe constituye un proyecto que ofrece la tecnología para el procesamiento y la visualización interactiva de vastos volúmenes de datos geospaciales en un globo virtual 3D. Es desarrollado principalmente por el Instituto de Ingeniería Geomática de la Universidad de Ciencias Aplicadas del Noroeste de Suiza. Puede tener varias categorías de datos de: imagen, elevación, puntos de interés, datos vectoriales y objetos 3D.

Esta herramienta ha sido utilizada para la visualización de edificios en 3D a partir de la base de datos de OpenStreetMap. La altitud correcta en el mundo virtual se obtiene a través del Modelo de Elevación Digital Global (ASTER GDEM), un servidor que suministra los parámetros de altura a partir de la captura de datos obtenidos por un satélite que registra las diferencias de temperatura a distintos niveles sobre la superficie terrestre. Ha sido utilizada no solo para la visualización de pequeñas porciones de territorio, sino de grandes ciudades como Nueva York y Beijing. Además trabaja con el navegador Internet Explorer en su versión 11 (SIEBER *et al.*, 2012).

Las soluciones Cesium y OpenWebGlobe tienen en común que utilizan WebGL para representar gráficos en 3D en la Web a partir de información geográfica almacenada, lo que constituye un aspecto a tener en cuenta en la búsqueda y construcción de la solución, basándose en que esta tecnología es libre y no necesita la utilización de plug-ins, de forma que puede ejecutarse en el navegador de cualquier usuario. Las soluciones mencionadas no son factibles de emplear debido a que cargan datos geográficos almacenados en servidores de internet con los cuales no se cuenta.

ReadyMap

ReadyMap es un conjunto de herramientas de código abierto de JavaScript para la prestación de un mundo en 3D a partir de datos de OpenLayers. Utiliza WebGL y el elemento Canvas de HTML5 para proyectar mapas 3D de alto rendimiento en el navegador sin necesidad de un *plug-ins*. Su librería JavaScript **ReadyMap SDK** permite agilizar la construcción de los mapas facilitando una respuesta rápida a las necesidades de los usuarios. Los componentes ReadyMap utilizan estándares abiertos. Funcionan bien juntos, pero también trabajan con la mayoría de los demás componentes de mapeo abierto, soportando protocolos como WMS, TMS, WFS y KML (ROMEU *et al.*, 2012).

WebGLEarth

WebGLEarth es una herramienta de código abierto que permite la visualización de mapas, imágenes de satélite y fotografías aéreas en la parte superior de un terreno virtual. Sus ventajas vienen dadas al posibilitar la visualización de las distintas regiones del mundo en cualquier navegador y plataforma, así como en dispositivos móviles. El proyecto se sustenta en el apoyo y la cooperación de la comunidad de desarrolladores. Constituye una solución basada en los principios de software gratuito disponible bajo la norma licencia GPLv3. Algunas de las características de esta solución se describen a continuación:

- La rotación y zoom del mundo, inclinación de la cámara, la libre circulación en el espacio.
- Se ejecuta sin un *plug-ins* en los navegadores web modernos.
- Muestra los mapas existentes de (OpenStreetMap).

Requisitos:

- Un navegador Web con soporte para HTML5 Canvas y WebGL extensión (Firefox 4, Chrome 9+, dispositivos móviles iPhone / iPad /).
- Lenguaje de programación: JavaScript(TECHNOLOGIES, 2013).

CityEngine

El visor web de CityEngine es una aplicación Web para visualizar escenas de ciudades en 3D y otras escenas 3D en un navegador. Se basa en la tecnología WebGL, que permite visualizar contenido en 3D en navegadores Web sin necesidad de instalar *plug-ins* adicionales. Este visor web permite interactuar con escenas de ciudades en 3D y otras escenas en 3D en un navegador para:

- Navegar por la escena desplazándose por la panorámica, ampliándola o reduciéndola y cambiando la perspectiva.
- Seleccionar capas específicas para visualizar.
- Deslizarse por la escena para revelar distintas propuestas y escenarios.
- Buscar entidades, atributos y metadatos en el contenido de la escena.

Requisitos:

Se puede usar el visor web de CityEngine en navegadores Web que admitan WebGL. Además, el hardware debe contar con una tarjeta gráfica instalada compatible con WebGL. Para obtener el mayor rendimiento, la tarjeta gráfica debe disponer de los controladores más recientes instalados y que haya al menos 512 MB de memoria de video. El visor no es apto para dispositivos móviles (WATSON *et al.*, 2008).

Las soluciones ReadyMap, WebGL Earth y CityEngine al igual que las dos anteriores utilizan WebGL, cuyas ventajas se describieron anteriormente, pero a su vez necesitan de propiedades de hardware y servicios de datos no disponibles hasta el momento, además de necesitar una conexión a internet para acceder a sus funcionalidades.

1.4 Conclusiones parciales.

La profundización en los fundamentos teóricos permitió encontrar los nexos existentes entre las diferentes categorías que comprende el objeto a investigar, reveló la necesidad de buscar alternativas para la representación de objetos en 3D, específicamente las edificaciones. Esto agregaría un valor significativo al SIG UCI, ofrecería un nivel más detallado, posibilidad de observación desde distintos ángulos, con texturas que simulen el entorno real, lo cual ayudaría a los usuarios a identificar con mayor facilidad los lugares a dirigirse dentro de la Universidad.

El análisis de la teoría precedente revela que son varias las soluciones para la modelación de objetos en 3D a partir de datos geográficos que evidencian el uso de la tecnología WebGL para la construcción de objetos 3D en la web y de librerías JavaScript, en función de agilizar este proceso. El empleo de esta tecnología permitiría eliminar sobre cargas del servidor, dado que emplea JavaScript, este es un lenguaje que ejecuta el código del lado del cliente, lo que permite que se ejecute parte de las funciones del sistema sin necesidad de realizar peticiones constantes al servidor.

Capítulo II: Tendencias y tecnologías actuales para el desarrollo del componente de visualización.

Para la construcción de un software es necesario definir una serie de aspectos de considerable importancia con el objetivo de que este pueda terminarse en tiempo y con la calidad deseada. El presente capítulo realiza una síntesis detallada de los elementos que permiten la construcción de un componente para visualizar edificios en 3D en la web, a partir de datos geográficos e integrarlo al SIG UCI, además se precisa la metodología de desarrollo de software y las herramientas a utilizar, así como la justificación de su elección.

2.1 Metodologías de desarrollo de software.

Una metodología de desarrollo de software en términos de ingeniería de software puede definirse como *“un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable”* (SOMMERVILLE, 2005). En la actualidad el creciente desarrollo de aplicaciones informáticas se encuentra vinculado al uso de metodologías de desarrollo de software, utilizadas estas como guía para estructurar, designar responsabilidades en el equipo de desarrollo, crear artefactos que permiten comprender mejor y agilizar el proceso de construcción.

2.1.1 El Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Unificado de Rational es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles (KRUCHTEN, 2004).

También puede definirse como: *“un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto”* (JACOBSON *et al.*, 2000).

RUP es un modelo en fases que identifica cuatro fases diferentes en el proceso de desarrollo de software, estas fases son:

Fase de inicio: El objetivo de la fase de inicio es el de establecer un caso de negocio para el sistema. Se deben identificar todas las entidades externas (personas y sistemas) que interactúan con el sistema y definir estas interacciones. Esta información se utiliza entonces para evaluar la aportación que el sistema hace al negocio. Si esta aportación es de poca importancia, se puede cancelar el proyecto después de esta fase.

Fase de elaboración: Desarrolla una comprensión del dominio del problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos clave del proyecto. Al terminar esta fase, se debe tener un modelo de los requerimientos del sistema (se especifican los casos de uso UML), una descripción arquitectónica y un plan de desarrollo de software.

Fase de construcción: Comprende el diseño del sistema, la programación y las pruebas. Durante esta fase se desarrollan y se integran las partes del sistema. Al terminar esta fase, debe tener un sistema, software operativo y la documentación correspondiente lista para entregarla a los usuarios.

Fase de transición: Se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad de usuario y hacerlo trabajar en un entorno real. Esto se deja de lado en la mayor parte de los modelos de procesos de software, pero es en realidad, una actividad de alto costo y a veces problemática.

Para la construcción del SIG UCI se empleó la metodología de desarrollo de software RUP, mencionada anteriormente, basándose en la experiencia acumulada a partir de la aplicación de esta metodología en la elaboración de la plataforma GeneSIG. Esta ha sido punto de partida para la creación de los SIG construidos en la Universidad.

A partir de que se pretende integrar un componente al SIG UCI que permita visualizar las edificaciones geo referenciadas de la Universidad en 3D, se decide utilizar esta metodología, aprovechando la experiencia que presentan el personal que labora en el proyecto en cuanto al uso de esta metodología, lo cual permite seguir la estructura de desarrollo de este sistema, esto facilitaría la construcción y elaboración del componente, así como de la documentación referente y de manera general aprovechar

todas las facilidades que brinda esta metodología. Su utilización posibilita una mayor comprensión del proceso de desarrollo de software, contribuye a que otros desarrolladores obtengan documentación que sirva como referencia para futuras investigaciones.

2.2 Lenguaje Unificado de Modelado (UML).

La utilización de los lenguajes de modelados toma un carácter importante en el desarrollo de software. Constituyen una herramienta que se emplea para construir los modelos del sistema. La metodología de desarrollo de software RUP emplea Lenguaje de Modelado Unificado (UML) como único lenguaje de modelado. Este es un lenguaje de modelado orientado a objetos, que permite representar gráficamente los elementos estáticos y dinámicos de una aplicación software.

A partir de que se elige RUP como metodología de desarrollo de software, se define la utilización de UML como lenguaje de modelado, basándose en las potencialidades de este, que permite la creación de distintos diagramas que agilizan el proceso de comprensión y construcción de la aplicación, lo cual facilita que el equipo del proyecto tenga un mayor dominio acerca de lo que se desea implementar.

2.3 Herramientas Case.

Las herramientas Case son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Estas herramientas tienen como objetivo: aumentar la productividad de las áreas de desarrollo y mantenimiento de los sistemas informáticos, reducir tiempos y costos de desarrollo y mantenimiento del software, mejorar la calidad del software desarrollado, optimizar la gestión y dominio sobre el proyecto en cuanto a su planificación, ejecución y control, así como perfeccionar el archivo de datos de conocimientos y sus facilidades de uso, reduciendo la dependencia de analistas y programadores (DÁVILA, 2013).

2.3.1 Herramientas Case Visual Paradigm for UML Enterprise Edition 8.0.

Es una herramienta case profesional de UML que admite el completo ciclo de vida del software. Permite la creación de todos los tipos de diagramas de clase, aplicar ingeniería inversa al código, generar código desde diagramas y generar documentación. Entre las razones que justifican la elección de la herramienta para ser utilizada en la investigación, se encuentran las siguientes:

- Tiene potencialidades para el trabajo con el lenguaje UML.
- Facilita y agiliza el proceso de desarrollo de software.
- Es multiplataforma.
- Posee una versión gratuita a diferencia de otras herramientas case.
- Permite generar informes para la documentación.

2.4 Lenguajes de programación.

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Para el desarrollo del componente de visualización en 3D de los edificios de la UCI, se utilizaron lenguajes de programación web, los cuales son descritos a continuación:

2.4.1 Preprocesador de Hipertexto (PHP) 5.

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma. Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente (PHP, 2006).

Este lenguaje es utilizado para establecer el acceso a la base de datos que contiene los datos geográficos de las edificaciones de la Universidad, capturarlos para su posterior procesamiento. El empleo de este lenguaje satisface las necesidades de adquisición de datos garantizando el uso de licencias libres, uno de los principios fundamentales para el desarrollo de aplicaciones en este centro.

2.4.2 JavaScript.

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (PÉREZ, 2009). Se emplea este lenguaje basándose en el principio de que la tecnología WebGL

establece un puente para el JavaScript que le permite usar la implementación nativa de OpenGL para la creación de gráficos 3D en entornos web.

2.5 Bibliotecas de clases para la representación tridimensional.

Las bibliotecas de clases constituyen un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca. Su fin es ser utilizada por otros programas independientes y de forma simultánea.

2.5.1 WebGL.

WebGL es una especificación estándar que se desarrolla actualmente para desplegar gráficos en 3D en navegadores web. Esta tecnología permite activar gráficos en 3D acelerados por hardware en páginas web, sin la necesidad de *plug-ins* en cualquier plataforma que soporte OpenGL 2.0 u OpenGL 2.0. Técnicamente es un enlace (binding) para JavaScript que le permite usar la implementación nativa de OpenGL 2.0.

Emplea el elemento Canvas de HTML5 y se accede mediante interfaces Document Object Model. La gestión de memoria automática se proporciona como parte del lenguaje JavaScript. Actualmente es soportado principalmente en Internet Explorer (versión 11), Google Chrome y Mozilla Firefox aunque también funciona con limitaciones en Opera browser y en Safari bajo el sistema operativo de Apple para computadoras de escritorio (CANTOR and JONES, 2012).

Se escoge esta tecnología para modelar edificios en 3D en la web en el SIG UCI, puesto que como se refería en las características de la misma, no es necesario la utilización de *plug-ins*, se encuentra embebida en los navegadores web, y sobre todo, los que son más utilizados por los usuarios de la Universidad: Google Chrome y Mozilla Firefox.

2.5.2 Bibliotecas de clases Three.js.

Es una biblioteca liviana escrita en JavaScript para crear y mostrar gráficos animados por ordenador en 3D en un navegador web y puede ser utilizada en conjunción con el elemento Canvas de HTML5, SVG o WebGL. Presenta diversas funcionalidades que permiten: la creación de escenas tridimensionales,

aplicación de efectos como sombra, iluminación, movimiento de cámaras, materiales y geometrías predefinidas, así como la inclusión o eliminación de objetos 3D en tiempo de ejecución. Además posee controles predeterminados para realizar movimientos de navegación en escenas tridimensionales (RAIDA and ZERBST, 2013).

La justificación de elección de esta librería se sustenta sobre la base de que posee numerosos métodos que permiten aplicar las funcionalidades de WebGL para la creación de gráficos 3D en la web de forma ágil y eficiente. El propósito general de esta librería no es la modelación de objetos 3D, a partir de datos geográficos, sin embargo es considerada entre las más potenciales en cuanto a funcionalidades, esta es una razón por la cual durante el desarrollo del trabajo se trata de aprovechar las ventajas que ofrece para la representación tridimensional de edificaciones a partir de datos geográficos. Además el análisis de otras librerías que permitan el trabajo con el WebGL permitió constatar que usan servicios de datos geográficos alojados en servidores con los cuales no se cuenta.

2.6 Framework de Desarrollo.

Un *framework* constituye una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, sobre la base de lo cual otro proyecto de software puede ser organizado y desarrollado (PATTON, 2005).

2.6.1 Framework de Java Script Ext JS 4.2.1.

ExtJS es una librería JavaScript que permite construir aplicaciones complejas, además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la potencialidad de crear interfaces de usuario bastante funcionales (ASHWORTH, 2012).

El empleo de este *framework* permite la creación de componentes de interfaz de usuario con un alto nivel de calidad, amigable e intuitivo que conduce a elevar el grado de aceptación del componente por parte de los usuarios finales. Otro de los factores que incide en el uso de este *framework* son las potencialidades que brinda para el trabajo con JavaScript orientado a objetos, lo cual facilita la creación de gráficos en 3D en entornos web.

2.7 Sistemas Gestores de Bases de Datos (SGBD) PostgreSQL 9.1.

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD, su código fuente se encuentra disponible libremente (GINESTÀ and MORA, 2012). Estas características favorecen su utilización, a lo que se le incorpora como un factor positivo que su extensión PostGIS soporta el trabajo con datos espaciales, lo cual favorece el almacenamiento y consulta de información geográfica.

2.7.1 PostGIS 1.5.8.

Ofrece soporte espacial a la popular base de datos objeto-relacional PostgreSQL. Puede ser usada como base de datos subyacente para sistemas de información geográfica (SIG) y aplicaciones de cartografía web. La administración de la base de datos es posible a través de PgAdmin y phpPgAdmin, entre otros. Garantiza importar y exportar datos mediante herramientas en línea de comandos (shp2pgsql, pgsq2shp, ogr2ogr, dxf2postgis) o a través de clientes SIG de escritorio o web. Además, estos clientes pueden crear, visualizar y manipular tablas espaciales (OBE and HSU, 2011). Los datos de ubicación geográfica de los edificios en el SIG UCI se encuentran almacenados en una base de datos en la que PostgreSQL es el sistema gestor de base de datos, con extensión Postgis, permitiendo almacenar los datos cartográficos en este sistema.

2.7.2 Sistema Administración de Bases de Datos pgAdmin III 1.14.0.

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración (OBE and HSU, 2011). Es empleada dicha herramienta por las facilidades que brinda para la administración de base de datos. Además de ser multiplataforma y de código abierto, existe experiencia en cuanto a su utilización.

2.8 Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado es una herramienta que realiza la integración de un conjunto de aplicaciones con el objetivo de crear otro software. Este tipo de sistema ofrece diversas ventajas al permitir automatizar tareas, empaquetar el código, compilarlo, ejecutarlo y generalmente depurarlo.

2.8.1 NetBeans 7.4.

NetBeans constituye un entorno de desarrollo gratuito, multiplataforma, y de código abierto. Permite el uso de un amplio rango de tecnologías para el desarrollo de aplicaciones de escritorio, web y dispositivos móviles. Da soporte a las siguientes tecnologías: Java, PHP, Groovy, C/C++, HTML5. Las características propias de esta herramienta de desarrollo descritas con anterioridad conducen a su utilización en la construcción del componente de visualización, realizando énfasis en las ventajas que ofrece para la programación en los lenguajes con que se realizará el mismo (PHP, JavaScript), permite el auto completamiento de código que facilita el trabajo del programador, además existe experiencia en el trabajo con este entorno de desarrollo.

2.9 Servidor Web Apache 2.2.

El servidor HTTP Apache es un servidor web de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual (KEW, 2008). Se emplea este servidor por las características descritas, realizando énfasis particularmente en sus propiedades que permiten operar en distintas plataformas, lo que posibilita el desarrollo del componente en distintos sistemas operativos, además de aplicar políticas de software libre, uno de los principios fundamentales adoptado por la Universidad.

2.10 Conclusiones parciales.

En el capítulo se abordaron las diferentes herramientas y tecnologías que se emplean para construir el componente de visualización en 3D para el SIG UCI. Se definió RUP como metodología de desarrollo de software, UML 8.0 como lenguaje de modelado, Visual Paradigm 5 como herramienta case, la librería three.js para la creación de edificios en 3D en el web, los lenguajes de programación PHP 5 y JavaScript,

NetBeans PHP 7.4 como entorno de desarrollo, el sistema gestor de base de datos PostgreSQL 9.1.2.1, su extensión para trabajar con datos geográficos Postgis 1.5.4.2, PgAdmin III para el manejo de la base de datos de PostgreSQL, como *framework* de desarrollo Ext JS 4.2.1. El proceso de elección de las herramientas y tecnologías descritas con anterioridad estuvo centrado en garantizar el cumplimiento de los principios básicos de uso tecnologías libres que favorezcan la construcción de la solución.

Capítulo III: Descripción de la solución propuesta.

3.1 Introducción:

En la actualidad la construcción de un software se sustenta sobre la base de la aplicación de una metodología de desarrollo de software que permita describir, agilizar y comprender este proceso. El presente capítulo sigue los principios del planteamiento anterior, en el mismo se realiza el análisis del modelo de dominio, la definición de requisitos funcionales y no funcionales, diagramas de casos de uso, que permiten comprender la propuesta de solución para el problema planteado.

3.2 Modelo del dominio.

La construcción de un software utilizando el lenguaje de modelado UML puede describirse a través del modelo de dominio o del modelo de negocio. Un modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software, sino de la propia realidad física.

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. De manera general, se puede definir un modelo de dominio como: *“una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés”* (PRESSMAN, 2005).

Los modelos de negocio en el proceso de desarrollo de un software permiten la comprensión del problema y su dominio, esto posibilita la identificación, análisis y especificación de los requisitos de software. Para elaborar un modelo de negocio es necesario tener el conocimiento de los procesos que actualmente serán establecidos en el nuevo sistema (PRESSMAN, 2005).

En este trabajo investigativo se decide utilizar el modelo de dominio debido a que no existe precisión en los procesos del negocio, al no contar con expertos en el negocio de la visualización en 3D de información

geográfica que puedan definir con exactitud los actores y casos de uso del negocio. De ahí que se emplee el modelo del dominio dado que la investigación se ha realizado mediante el análisis de sistemas que tengan un funcionamiento similar.

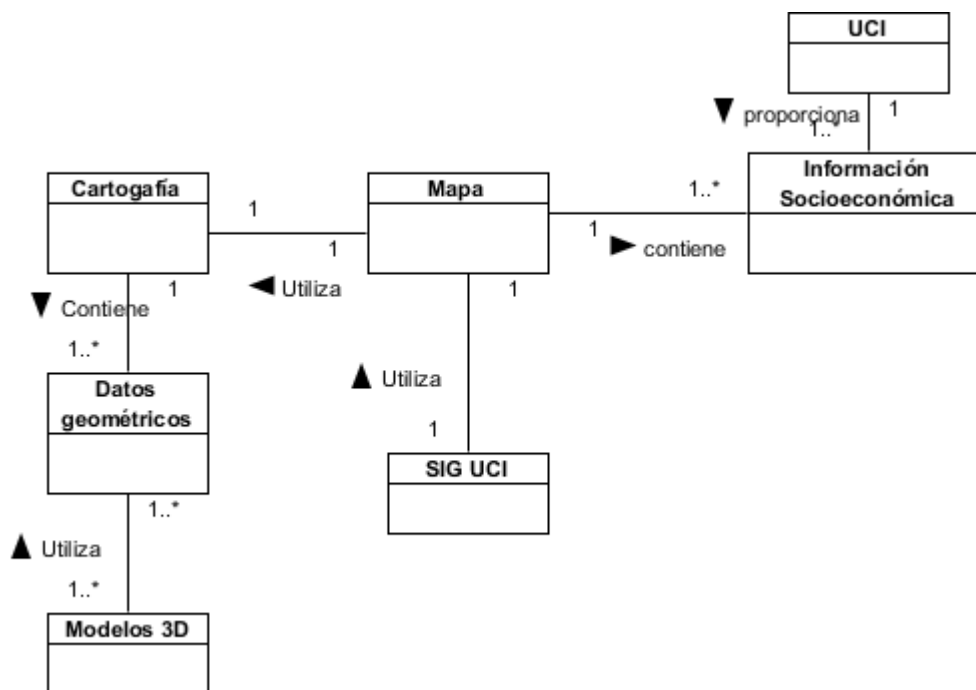


Fig. 1: Diagrama del modelo de dominio.

Descripción de las clases del dominio del Modelo de Dominio.

UCI: Universidad de las Ciencias Informáticas, centro en el cual se solicita información referente a las distintas edificaciones y personal existente dentro de la misma a partir del empleo de un mapa.

Información Socioeconómica: Conjunto de datos referentes a aspectos sociales y económicos de una región determinada, particularmente de la UCI.

Mapa: Proyección bidimensional de la superficie terrestre, que establece una proporción entre el territorio real y la porción de territorio que este representa.

Cartografía: Se refiere al proceso de elaboración de mapas geográficos, mediante el empleo de técnicas de medición que permiten determinar las coordenadas geográficas del terreno.

Datos Geométricos: Aquellos datos que permiten la elaboración de figuras planas, ejemplo: polígonos.

Modelos 3D: Conjunto de datos geométricos que permiten realizar la modelación tridimensional de un objeto.

SIG UCI: Sistema de Información Geográfico de la Universidad de las Ciencias Informáticas, que muestra la información socioeconómica de la misma.

3.2.1 Descripción del diagrama del modelo dominio.

La Universidad de las Ciencias Informáticas proporciona información socioeconómica referente a las distintas áreas ubicadas en el territorio que esta comprende, dicha información se encuentra contenida en el mapa que emplea el SIG UCI, lo cual permite al sistema ofrecerla a todo el personal que interactúe con este. El mapa utiliza una cartografía que permite obtener la localización de distintos lugares en la Universidad, la cartografía a su vez presenta datos geométricos bidimensionales que representan los polígonos de los edificios, los cuales son empleados para la construcción de modelos tridimensionales de los edificios, que posteriormente el SIG UCI ofrecerá la posibilidad de ser visualizados.

3.3 Requisitos de software.

Los requerimientos o requisitos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. Para una mejor comprensión del sistema se establecen dos tipos de requisitos fundamentales: funcionales y no funcionales.

3.3.1 Técnica de extracción de requisitos.

La determinación de los requisitos de un sistema se encuentra estrechamente vinculada a la aplicación de diversas técnicas de obtención de requisitos, las cuales permiten, a partir de la comprensión de las necesidades del usuario, definir lo que el sistema debe hacer y cómo debe funcionar. Enmarcados en el ámbito del problema identificado se empleó la entrevista como técnica de obtención de requisitos, la cual

fue de gran utilidad para adquirir información a través del intercambio directo entre distintos usuarios de la Universidad, así como los integrantes del proyecto **Aplicativo SIG**, mediante el cual se captaron las distintas opiniones e ideas que permiten comprender el proceder para establecer una solución que sea viable.

3.3.2 Requisitos funcionales.

Expresan la naturaleza del funcionamiento del sistema, es decir, cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento. Son ejemplos de requerimientos funcionales los cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, deban cumplirse en un sistema.

RF.1: Visualizar objeto tridimensional en colores: El sistema debe ser capaz de modelar (construir) los objetos tridimensionales de tipo edificio a partir de los datos geométricos almacenados.

RF.2: Visualizar objetos vectoriales 3D: El sistema debe ser capaz de visualizar en modelo de malla el objeto tridimensional de tipo edificio en el SIG UCI.

RF.3: Visualizar el objeto desde diferentes vistas: El sistema debe ser capaz de rotar el objeto tridimensional de tipo edificio permitiendo visualizarlo desde todos sus ángulos.

RF.4: Acercar vista del objeto: El sistema debe ser capaz de acercar la vista del objeto tridimensional de tipo edificio.

RF.5: Alejar vista del objeto: El sistema debe ser capaz de alejar la vista del objeto tridimensional de tipo edificio.

3.3.3 Requisitos no funcionales.

Este tipo de requisito incluye todo lo del sistema con excepción de la funcionalidad. Toma en cuenta las restricciones sobre el espacio de posibles soluciones, estos requisitos “representan aquellos atributos que debe exhibir el sistema pero que no constituyen una funcionalidad específica. Por ejemplo requisitos de usabilidad, fiabilidad, eficiencia, portabilidad” (JACOBSON *et al.*, 2000).

La descripción de los requisitos no funcionales del componente de visualización en 3D de las edificaciones para el SIG UCI se muestra a continuación:

Usabilidad:

- El componente creado para la visualización en 3D puede ser usado por los usuarios con conocimientos básicos de informática.
- Debe tener una interfaz visual sencilla, de ahí que se utilice el mouse para las funciones de visualización, dígase hacer zoom al edificio indicado.
- Debe contar con una serie de botones que ejecuten las funciones de visualización, de forma que los usuarios no experimentados puedan reconocer el significado de estos botones y facilitar la interacción con el componente.

Rendimiento:

El tiempo de respuesta del sistema depende de la cantidad de edificios a visualizar teniendo en cuenta que la visualización en 3D se basa en el empleo de gráficos acelerados por hardware.

Fiabilidad:

- La disponibilidad será permanente para las personas de la Universidad.
- Debe garantizar el soporte para recuperación ante errores.

Requisitos de hardware:

- Las computadoras clientes deben contar con tarjeta de red.
- Deben poseer al menos 128 MB de RAM.
- Necesita como mínimo 10 GB de disco duro.
- Requiere al menos 128 MB de video.

Requisitos de Software:

El componente web de visualización debe garantizar una mayor portabilidad, al eliminar la mayor cantidad de dependencias, requiere funcionar en los principales navegadores que son empleados por los usuarios de la Universidad: Google Chrome y Mozilla Firefox.

Restricción de implementación:

- El sistema debe implementarse con la utilización de los estándares de codificación de forma que se establezca una coordinación en el código que permita su entendimiento.

3.4 Descripción del sistema propuesto.

A continuación se realiza la descripción del sistema propuesto a partir de los artefactos que genera la metodología de desarrollo de software RUP:

3.4.1 Descripción de los actores del sistema.

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que se está construyendo de la misma forma (JACOBSON *et al.*, 2000).

Tabla 1: Descripción de los actores del sistema

Actor	Descripción
Usuario	Personal dentro de la Universidad que necesite localizar un edificio.

3.4.2 Diagrama de Casos de Uso del Sistema.

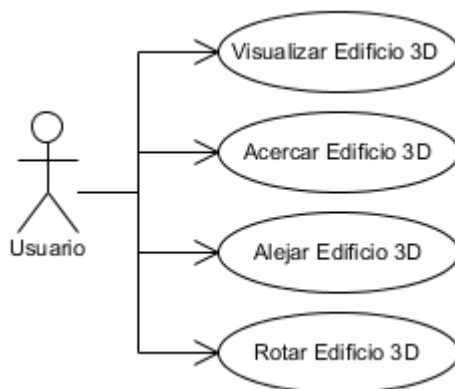


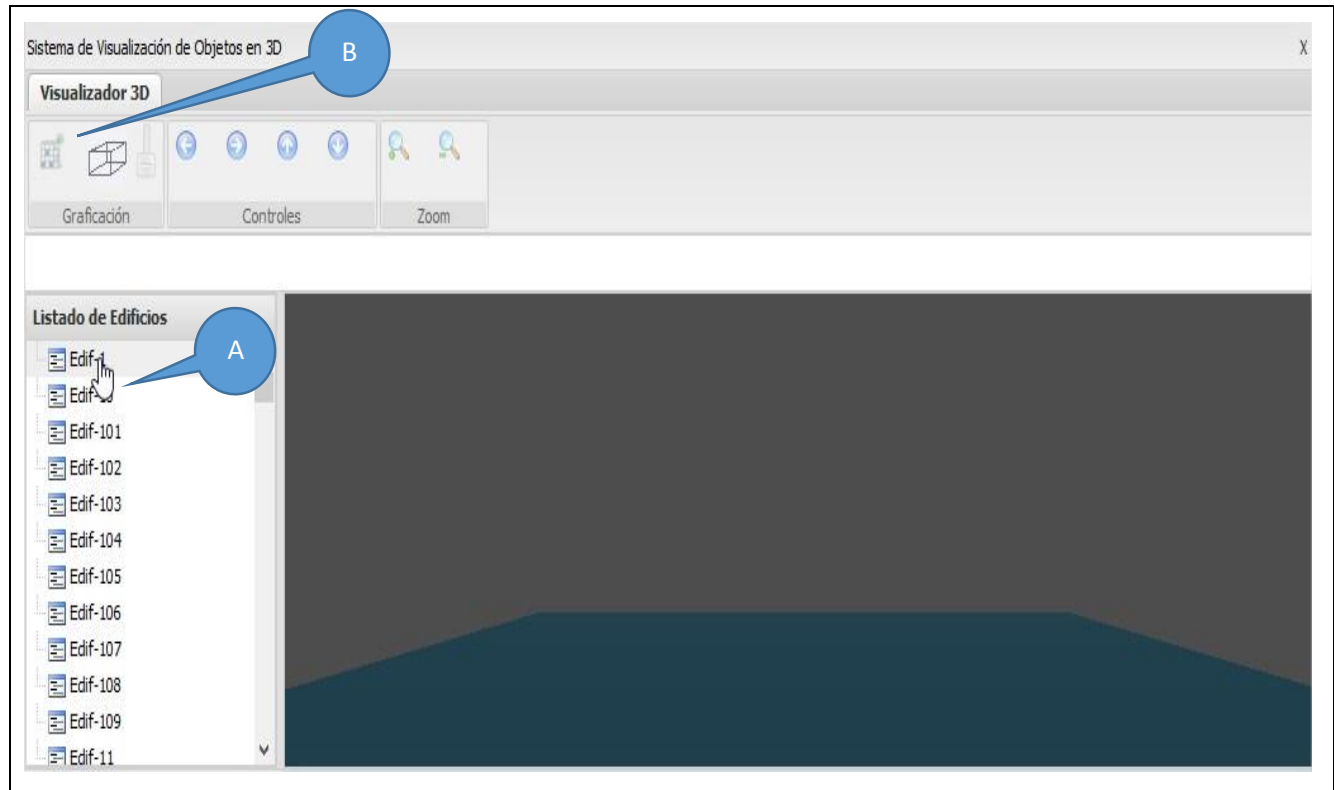
Fig. 2: Diagrama de Casos de Uso del Sistema.

3.4.3 Descripción textual de los Casos de Uso del Sistema.

Tabla 2: Descripción del caso de uso visualizar edificio 3D.

Caso de Uso:	Visualizar Edificio 3D
Actores:	Usuario
Propósito	Se pretende que el usuario sea capaz de apreciar visualmente el edificio en 3D de acuerdo a la forma que tiene almacenada en la base de datos.
Resumen:	El caso de uso se inicia cuando el usuario selecciona el edificio que quiere visualizar y elige la opción modelar en 3D, y finaliza cuando el sistema visualiza el edificio.
Precondiciones:	El usuario debe haber seleccionado el edificio que quiere visualizar.
Referencias	RF 1, RF 2.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

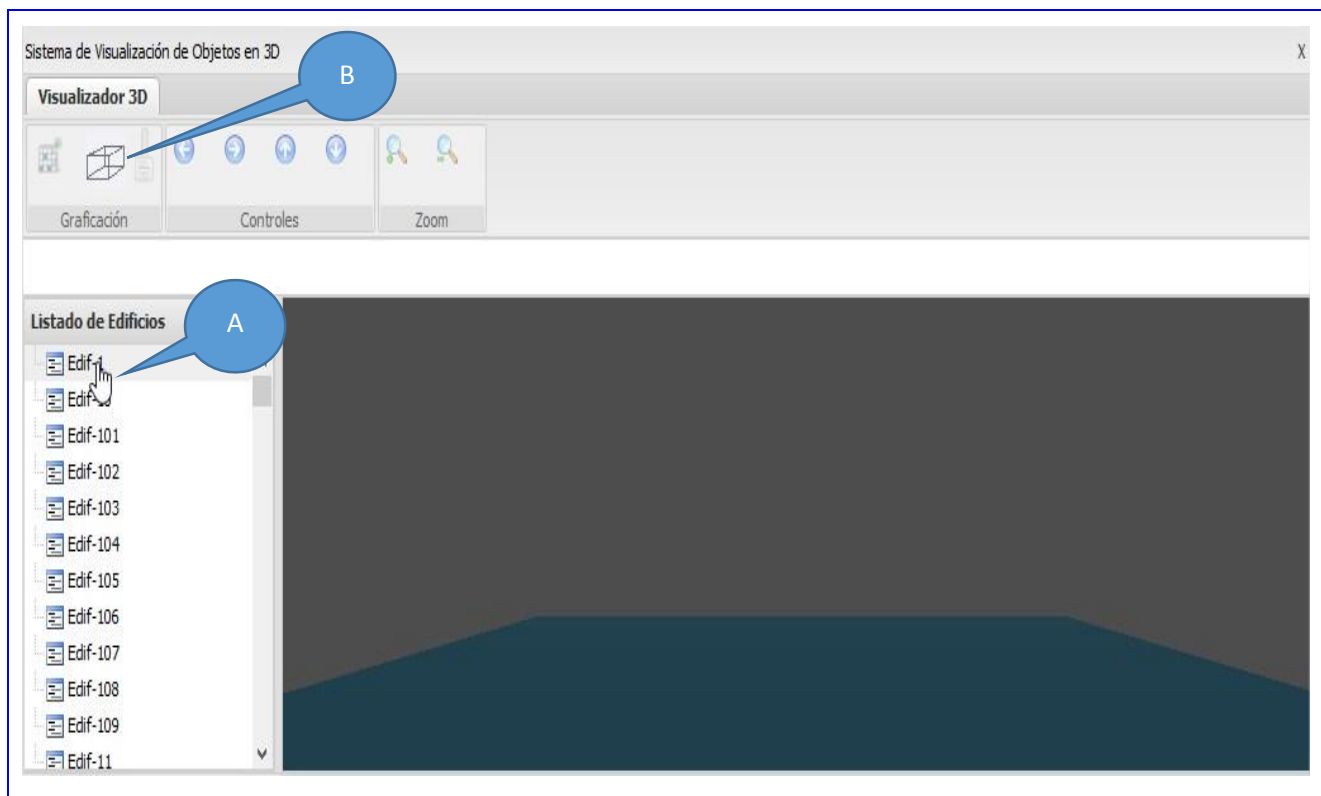
1. El usuario accede al SIG UCI.	2. El sistema muestra la interfaz con el mapa de la Universidad.
3. El usuario selecciona la opción modelar 3D.	4. El sistema muestra la interfaz de representación de edificios 3D
<p>5. El usuario elige el edificio que quiere visualizar en 3D y posteriormente presiona uno de los botones de representación tridimensional.</p> <p>Si el usuario desea visualizar el edificio a partir de la representación por colores vea la sección: Representar edificios a partir de la visualización por colores.</p> <p>En caso de que el usuario desee visualizar el edificio a partir de la representación en un modelo de malla, vea la sección: Representar edificios aplicando modelo de malla.</p>	6. El sistema muestra en el panel de visualización la representación tridimensional del edificio, en dependencia del tipo de visualización que seleccione el usuario, finalizando de esta forma el caso de uso.
Flujo Normal de Eventos Sección: Representar edificios a partir de la visualización por colores.	
Acción del Actor	Respuesta del Sistema
1. El usuario accede al SIG UCI.	2. El sistema muestra la interfaz con el mapa de la Universidad.
3. El usuario selecciona la opción modelar 3D.	4. El sistema muestra la interfaz de representación de edificios 3D
5. El usuario elige el edificio que quiere visualizar en 3D y posteriormente presiona el botón de representación tridimensional.	6. El sistema muestra en el panel de visualización la representación tridimensional del edificio.
Prototipo de Interfaz	



Flujo Normal de Eventos Sección: Representar edificios aplicando modelo de malla.

Acción del Actor	Respuesta del Sistema
1. El usuario accede al SIG UCI.	2. El sistema muestra la interfaz con el mapa de la Universidad.
3. El usuario selecciona la opción modelar 3D.	4. El sistema muestra la interfaz de representación de edificios 3D
5. El usuario elige el edificio que quiere visualizar en 3D y posteriormente presiona el botón de representación tridimensional en modelo de malla.	6. El sistema muestra en el panel de visualización la representación tridimensional del edificio.

Prototipo de Interfaz



3.5 Conclusiones parciales.

La descripción del modelo de dominio del sistema permitió realizar una descripción del marco actual del problema, al identificar las principales clases del dominio del problema. Se identificaron los requisitos funcionales que el sistema debe realizar, estos a su vez, generaron cuatro casos de uso que guían el desarrollo del componente. Además se identificaron los requisitos no funcionales que debe cumplir el componente para que este sea aplicable de acuerdo con las condiciones y marco de trabajo en que se va a utilizar. De forma general quedaron sentadas las bases para dar comienzo a la implementación.

Capítulo IV: Construcción de la solución propuesta.

4.1 Introducción.

En el presente capítulo se define la estructura arquitectónica, se reflejan los estilos y patrones a partir de los cuales se desarrollará el componente. A través de los distintos artefactos que genera la metodología RUP se describe el flujo de implementación, lo cual permite facilitar la comprensión de este proceso. Con posterioridad se llevan a cabo las pruebas al componente para comprobar el correcto funcionamiento de los requisitos funcionales, así como la corrección de los errores detectados.

4.2 Arquitectura del software.

“La arquitectura de software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos” (BASS, 2007).

La utilización de arquitecturas de software permite estructurar el software, influye en su correcta construcción, esto a su vez contribuye a elevar su calidad, sin lugar a dudas es necesario su empleo para la comprensión del sistema, organización del desarrollo, impulsar la reutilización y de forma general hacer evolucionar el sistema. Diversos autores se refieren a la arquitectura de software como dos vertientes diferentes denominadas: estilos y patrones arquitectónicos, mientras que otros hacen referencia a esta únicamente como patrones arquitectónicos. El desarrollo de esta investigación se sustentó en el criterio de separación entre los estilos y los patrones arquitectónicos.

4.2.1 Estilos Arquitectónicos.

Los estilos arquitectónicos se definen como: *“una familia de sistemas de software en términos de un patrón de organización estructural, que define un vocabulario de componentes y tipos de conectores y un conjunto de restricciones de cómo pueden ser combinadas. Para muchos estilos puede existir uno o más modelos semánticos que especifiquen cómo determinar las propiedades generales del sistema partiendo de las propiedades de sus partes” (SHAW and GARLAN, 1996).*

Durante el desarrollo del componente de visualización en 3D se evidenció el empleo del estilo arquitectónico llamada y retorno. En este estilo el sistema se constituye de un programa principal que tiene el control del sistema y varios subprogramas que se comunican con éste mediante el uso de llamadas (BASS, 2007). El estilo llamada y retorno permite construir una estructura de programa relativamente fácil de modificar y ajustar a escala.

4.2.2 Patrones Arquitectónicos.

“Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Son plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una aplicación con amplitud de todo el sistema y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software” (BUSCHMANN et al., 2001). Diversos patrones fueron empleados para la realización del componente de visualización, entre estos se encuentran:

Patrón arquitectónico en capas:

El estilo arquitectural en capas se basa en una distribución jerárquica de los roles y responsabilidades para proporcionar una división efectiva de los problemas a resolver (DE LA TORRE et al., 2010). Para estructurar el componente de visualización se empleó específicamente el patrón arquitectónico en tres capas, el cual permite organizar el modelo de diseño en tres capas lógicas distintas, cada una con un grupo de interfaces bien definidas y en las que una capa solo muestra relación directa con la capa inmediatamente superior o inferior. La descripción de las capas se realiza a continuación:

Capa de presentación: Encargada de la interacción entre el usuario y el sistema, esta ofrece un panorama visual del sistema al usuario, le muestra información y obtiene información de este en un mínimo de proceso. Se comunica únicamente con la capa de negocio.

Capa de negocio: En esta capa se establecen todas las reglas que debe cumplir el sistema, es la encargada de manejar las peticiones del usuario, procesar la información y enviar respuestas una vez

concluido el procesamiento. Realiza la comunicación con la capa de presentación para obtener las solicitudes y enviar respuestas a estas, y con la capa de acceso a datos para obtener o recuperar los datos contenidos en base de datos.

Capa de acceso a datos: Es la capa encargada de almacenar los datos del sistema y de los usuarios, realiza el almacenamiento y devolución de los datos a la capa de negocio, es la única que puede tener acceso a los datos.

Arquitectura orientada a objetos:

“Este patrón define el sistema como un conjunto de objetos que cooperan entre sí como un conjunto de procedimientos. Los objetos son discretos, independientes y poco acoplados, se comunican mediante interfaces y permiten enviar y recibir mensajes” (DE LA TORRE et al., 2010). El patrón en cuestión describe el uso de objetos que contienen los datos y el comportamiento para trabajar, además tienen un rol o responsabilidad distinta. Hace hincapié en la reutilización a través de la encapsulación, la modularidad, el polimorfismo y la herencia.

Arquitecturas basadas en componentes:

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado (VIGNAGA and PEROVICH, 2003).

4.3 Patrones de Diseño.

“Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular” (BUSCHMANN et al., 2001).

Otros autores expresan su opinión respecto a los patrones de diseño:

“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma” (GAMMA et al., 1994).

Ambos conceptos expresan la importancia de la utilización de un patrón de diseño para la construcción de un software, al permitir su estructuración, reutilización, acoplamiento y correcto funcionamiento. Existen diversas agrupaciones de patrones de diseño, el desarrollo de esta investigación evidenció el uso de alguno de los principales.

4.3.1 Patrones de Diseño GRASP.

GRASP es el acrónimo de General Responsibility Assignment Software Patterns, estos tipo de patrones describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones, estos principios adquieren importancia para diseñar eficientemente un software (LARMAN, 1999). Entre los patrones GRASP utilizados durante la construcción de la aplicación se pueden destacar:

Patrón Experto:

Este patrón plantea la asignación de una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos. El empleo de este patrón queda evidenciado en la clase building del paquete cliente, la cual posee la información necesaria para la creación de objetos de tipo edificio a través del método build3D, esta constituye su principal responsabilidad.

Patrón Creador:

El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se

da soporte al bajo acoplamiento. Dicho patrón se encuentra reflejado en la clase StructureSet del paquete cliente, encargada de crear instancias de la clase THREEDEStructure, con el objetivo de representar un conjunto de estructuras 3D.

Patrón Bajo Acoplamiento:

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Este patrón plantea la necesidad de eliminar la mayor cantidad de dependencias entre las clases, de esta forma se reduce el impacto de los cambios y se permite una mayor reutilización de estas. Dicho patrón se encuentra aplicado en el mayor número de clases en interés de eliminar las dependencias entre clases.

Patrón Alta Cohesión:

La cohesión es una medida de la forma en que están relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. El empleo de este patrón aumenta el nivel de claridad y facilidad con que se entiende el diseño, permite una mayor reutilización y simplifica el mantenimiento y las mejoras de funcionalidad. Es utilizado este patrón en el mayor número posible de clases para acotar el trabajo realizado por aquellas que posean responsabilidades que se encuentren relacionadas entre sí.

Patrón Controlador:

Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. El empleo de este patrón se ve reflejado en la clase V3D del paquete cliente, encargada de manejar las peticiones que realiza el usuario y las respuestas que el sistema debe ofrecer (LARMAN, 1999).

Existe otro grupo de patrones de diseño: los patrones GOF, los cuales constan de un conjunto de 23 patrones de diseño divididos en 3 categorías: creacionales, estructurales y de comportamiento, algunos de estos fueron utilizados en la construcción del componente, los cuales se describen a continuación:

4.3.2 Patrones de diseño GoF.

Los patrones de diseño GoF (*Gang of Four*) conforman un conjunto de 23 patrones de diseño agrupados en las categorías: creacionales, estructurales y de comportamiento, cuyo objetivo es describir soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. (GAMMA *et al.*, 1994) A continuación se ejemplifican los que fueron empleados para la construcción de la solución:

Singleton:

Este patrón garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El empleo de este patrón se encuentra evidenciado en la clase Utils, en la que solo se crea una instancia de esta para acceder a los métodos que proporciona, no es necesario crear una instancia de la clase cada vez que se requiera emplear alguno de sus métodos.

Decorator:

El patrón Decorator responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto, lo cual permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Se refleja el uso de este patrón en las clases Vcollection y StructureSet.

4.4 Modelo de diseño.

*“El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen un impacto en el sistema a considerar” (JACOBSON *et al.*, 2000).*

La aplicación de una arquitectura de software derivada en una serie de estilos y patrones arquitectónicos define el diseño estructural que tendrá el sistema, posibilita comprender el proceso de desarrollo del software, lo que permite llevar a cabo la construcción de este, de forma que se logren los objetivos propuestos y se elabore un producto con la calidad y organización deseada.

4.4.1 Diagrama de Clases del Diseño (DCD).

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (LARMAN, 1999).

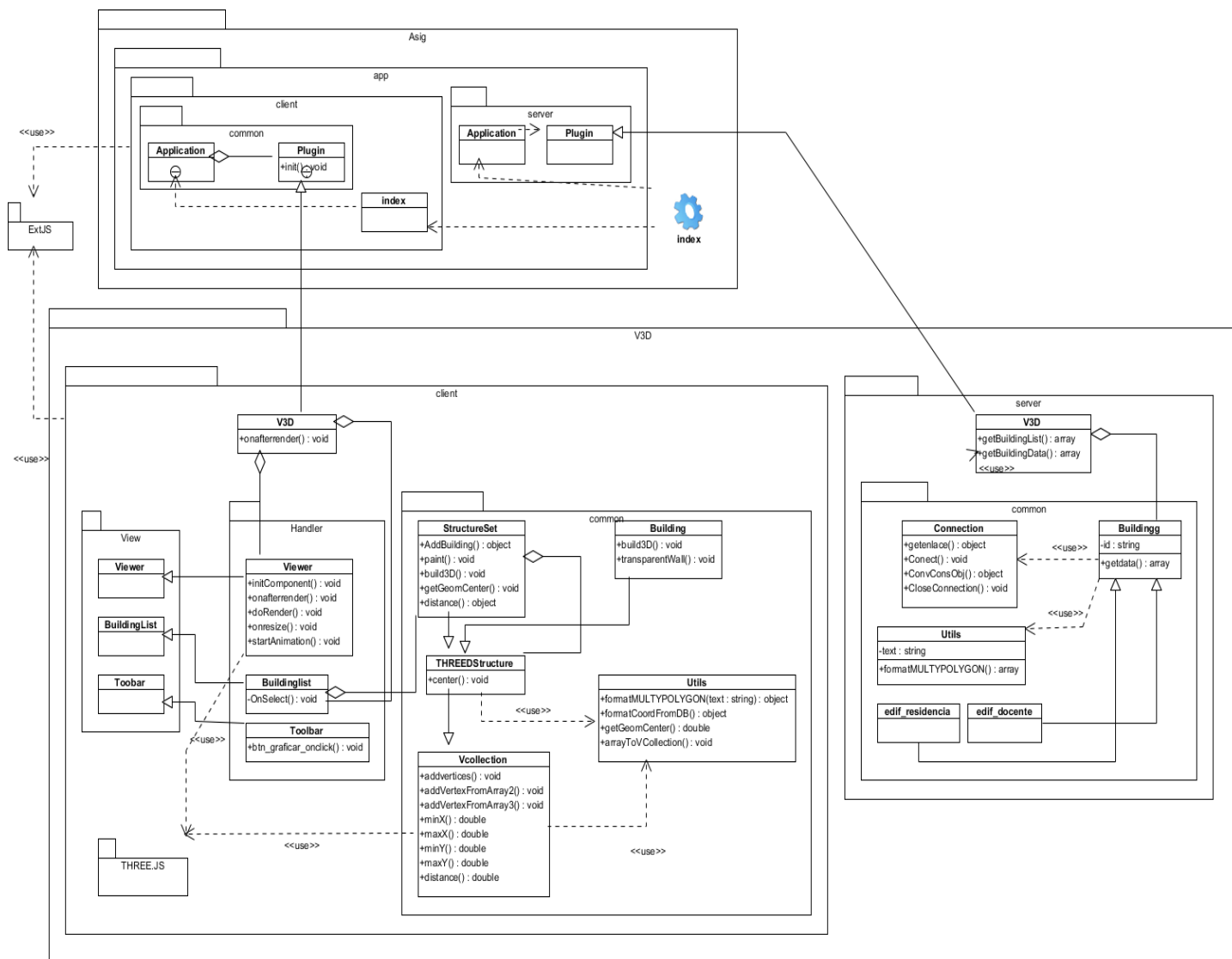


Fig. 3: Diagrama de clases del diseño.

4.5 Diseño de la Base de Datos

El diseño de la base de datos representa uno de los elementos fundamentales para la elaboración de un sistema, puesto que estructura correctamente el acceso a datos, con el objetivo de satisfacer la necesidad de obtención de reportes sobre estos datos. Para el componente de visualización se empleó un diseño de base de datos existente el cual cuenta con los datos geográficos de los edificios de la Universidad.

4.5.1 Modelo entidad-relación.

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades (LARMAN, 1999).

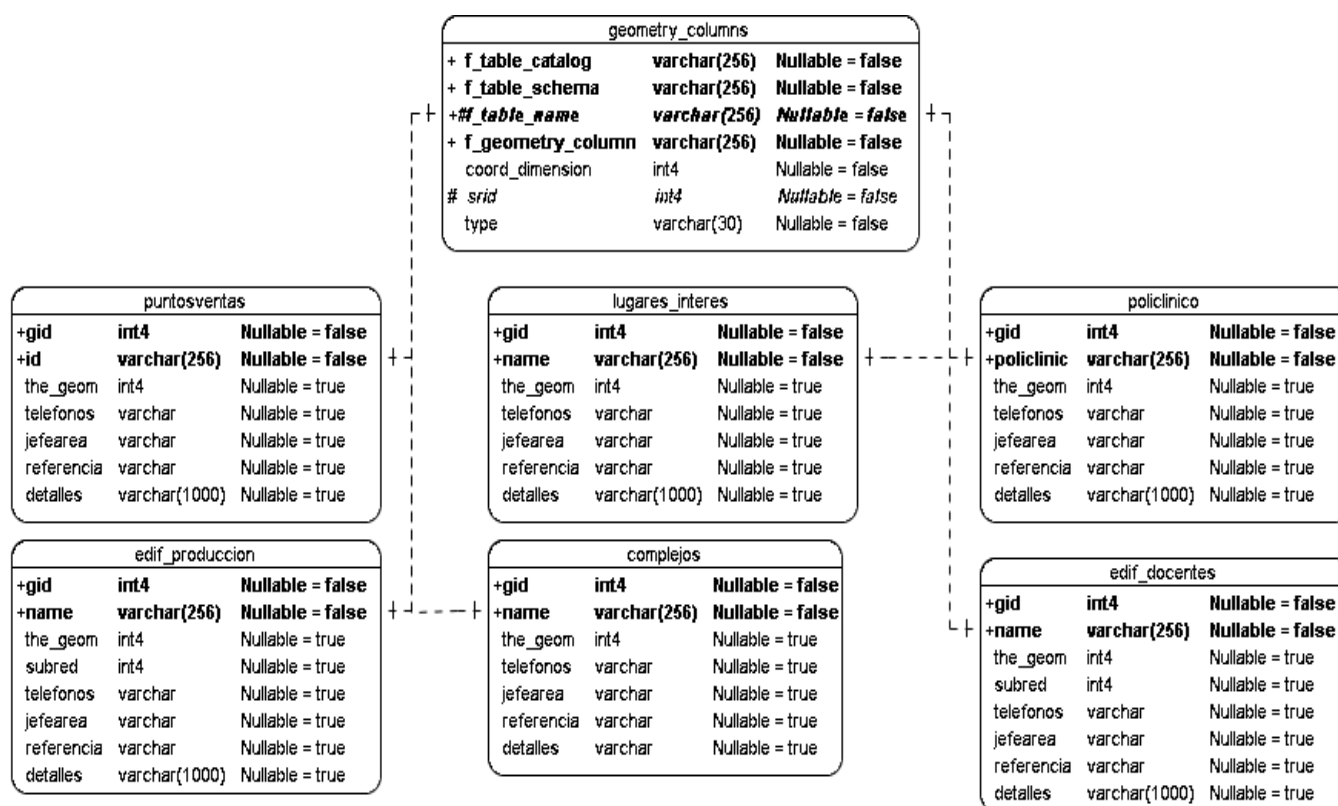


Fig. 4: Modelo entidad relación parte 1.

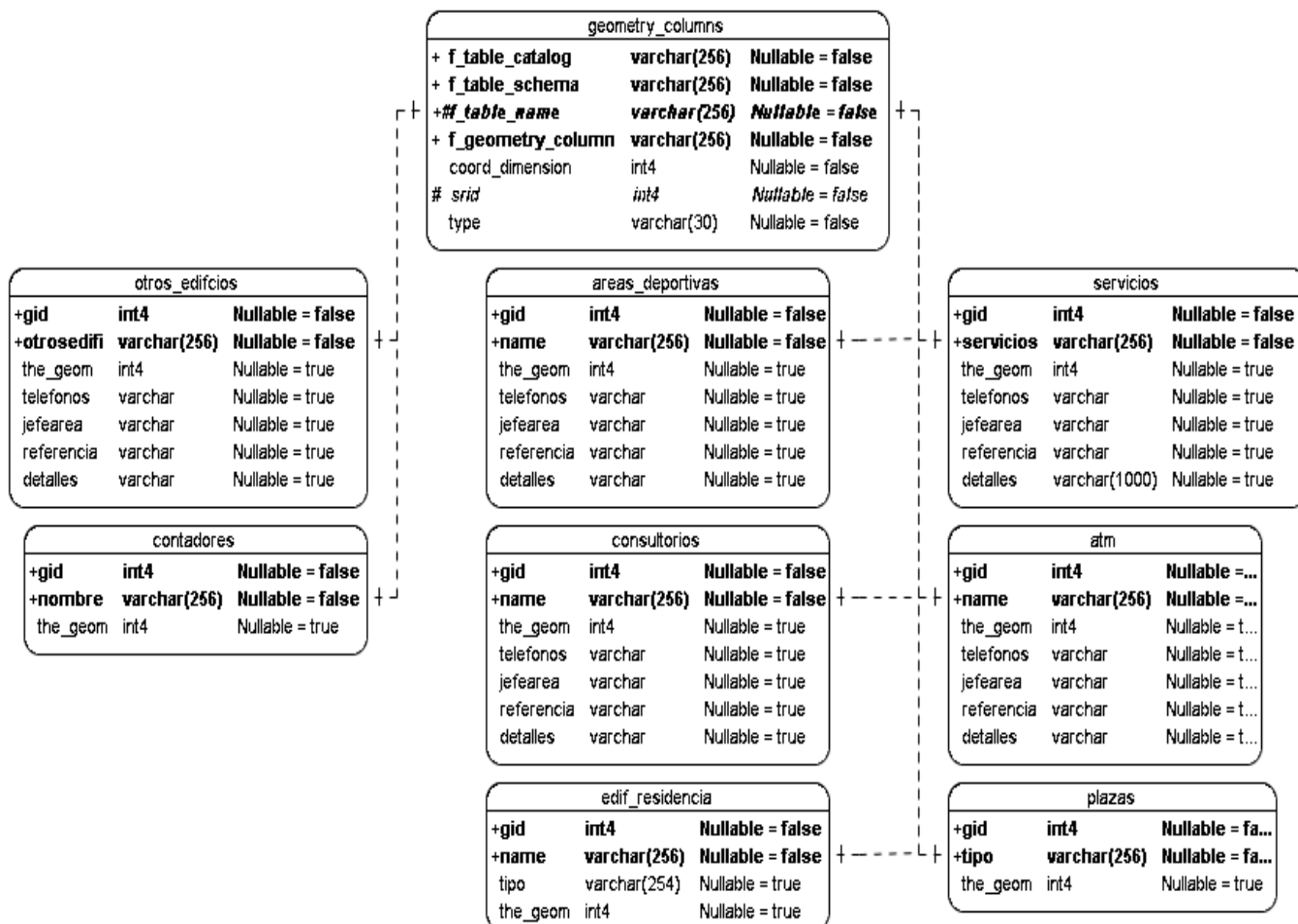


Fig. 5: Modelo entidad relación parte 2.

4.6 Modelo de Implementación.

El modelo de implementación es una correspondencia directa de los modelos de diseño y de despliegue, describe cómo los elementos del modelo de diseño se implementan en términos de componentes. También describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y de cómo dependen los componentes unos de otros (JACOBSON *et al.*, 2000).

4.6.1 Diagrama de Componentes.

La implementación de un sistema puede reflejarse a través del empaquetamiento físico de los elementos del modelo de diseño del mismo, de forma que se permita organizar esta implementación. El presente diagrama describe la estructura que posee el componente de visualización:

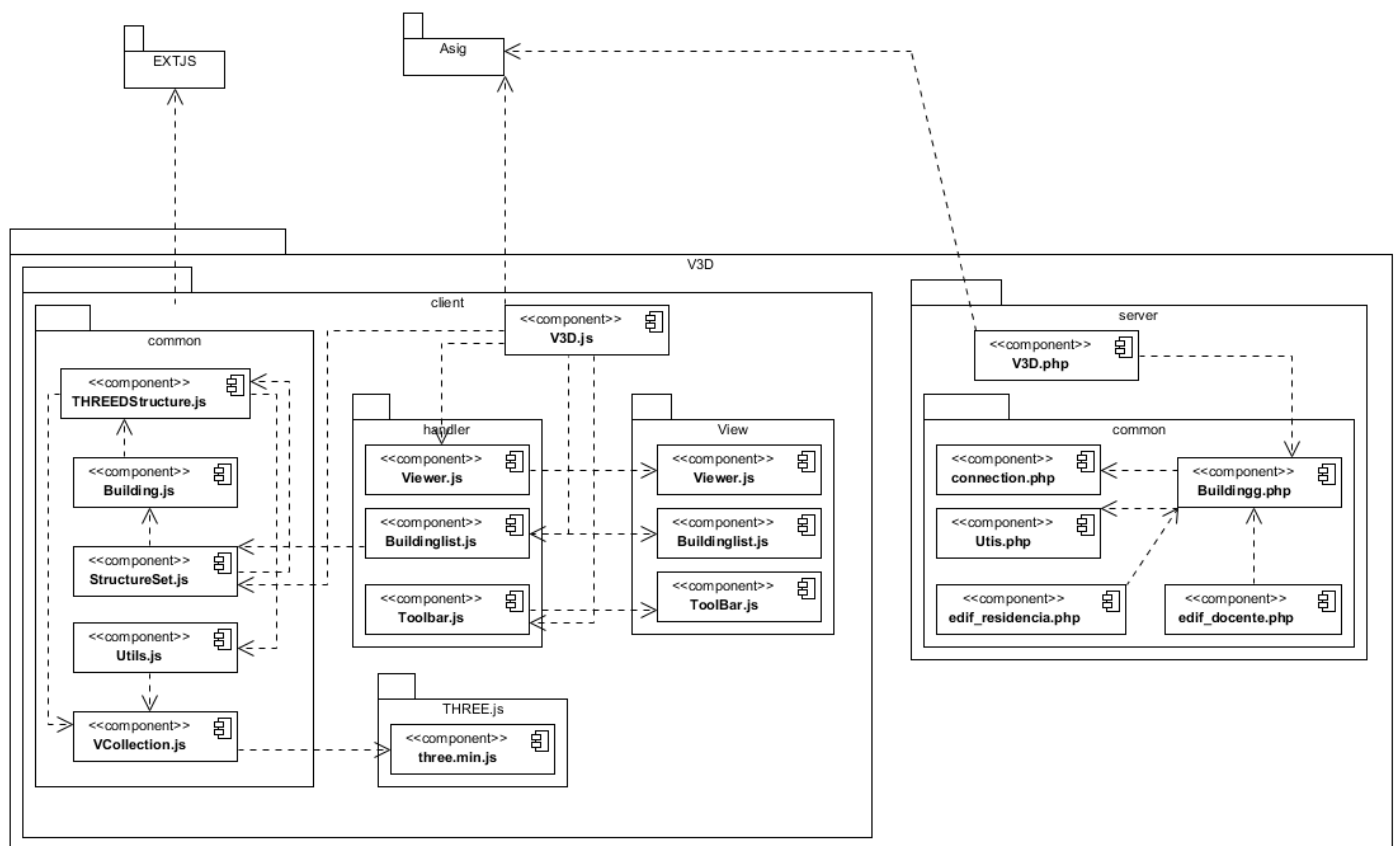


Fig. 6: Diagrama de componentes.

4.7 Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (JACOBSON *et al.*, 2000).

El componente de visualización se realiza en un entorno web, sobre el cual se emplean los protocolos de comunicación TCP/IP y HTTP:

Protocolo de Transferencia de Hipertexto (HTTP): Permite transferir información entre un cliente o un navegador web y un servidor web. Está basado en el modelo cliente servidor, en donde un cliente HTTP abre una conexión y realiza una solicitud al servidor, este a su vez envía una respuesta a la petición y finalmente cierra la conexión (CASAMOR, 2003).

Protocolo de Control de Transmisión/Protocolo de internet (TCP/IP): Constituye la base de Internet, y es empleado para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN). Este protocolo posee un alto grado de fiabilidad, su uso es adecuado para redes grandes y medianas, así como redes empresariales y es compatible con las herramientas estándar para analizar el funcionamiento de la red (CASAMOR, 2003).

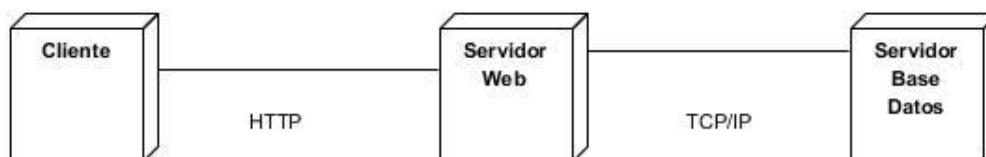


Fig. 7: Diagrama de despliegue.

4.8 Pruebas de software.

Las pruebas de software constituyen el proceso que permite medir el grado en el que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba. Este proceso adquiere una importancia significativa al permitir: comprobar el correcto funcionamiento del sistema, identificar deficiencias en software construido que luego deben ser corregidas para obtener un producto de calidad que cumpla las expectativas de los usuarios.

El proceso de desarrollo de pruebas de un software es guiado a través de dos perspectivas diferentes: sobre la lógica interna del programa, que puede ser comprobada mediante el uso de técnicas de diseño de casos de uso de pruebas de caja blanca, y sobre la perspectiva de cumplimiento de los requisitos del software, mediante técnicas de diseño de casos de prueba de caja negra.

4.8.1 Pruebas de Caja Negra.

Este tipo de pruebas fue aplicado sobre el componente de visualización con el objetivo de comprobar el correcto funcionamiento de los requisitos funcionales. Para la puesta en práctica de este tipo de pruebas se empleó la técnica partición equivalente, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada.

Se selecciona como parte del proceso de desarrollo de software el diseño del caso de prueba correspondiente al caso de uso visualizar edificio 3D.

Tabla 3: Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Representar edificios a partir de la visualización por colores.	EC 1.1 Representar satisfactoriamente edificio 3D.	Permite realizar la representación visual de los objetos de tipo edificio por colores.	<ol style="list-style-type: none"> 1. El usuario accede al SIG UCI y selecciona la opción de representación 3D. 2. El sistema carga la interfaz del visualizador, sobre el área izquierda de este aparece la lista de los edificios. 3. El usuario realiza la selección del edificio que quiere representar, presiona el botón de modelación tridimensional. 4. El sistema muestra en el panel de visualización el edificio 3D.
SC2: Representar edificios	EC 2.1 Representar satisfactoriamente	Permite realizar la representación visual de los objetos de tipo edificio	<ol style="list-style-type: none"> 1. El usuario accede al SIG UCI y selecciona la opción de representación

aplicando modelo de malla	edificio 3D.	aplicando un modelo de malla a partir datos almacenados	3D. 2. El sistema carga la interfaz del visualizador, sobre el área izquierda de este aparece la lista de los edificios. 3. El usuario realiza la selección del edificio que quiere representar, presiona el botón de graficado en modelo de malla. 4. El sistema muestra en el panel de visualización el edificio 3D.
---------------------------	--------------	---	---

Tabla 4: Descripción de las variables de entrada

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	criterio de selección	Árbol de selección	No	Se debe seleccionar el edificio que se quiera representar en 3D.

SC1: Representar edificios aplicando representación por colores.

Tabla 5: Matriz de datos SC1

ID del EC	Escenario	V1	Respuesta del sistema	Resultado de la prueba
EC 1.1	EC 1.1 Representar satisfactoriamente edificio 3D.	V Edificio 106	El sistema muestra la representación visual de los objetos de tipo edificio aplicando un color específico partir de los datos almacenados.	Satisfactoria

SC2: Representar edificios aplicando modelo de malla.

Tabla 6: Matriz de Datos SC2

ID del EC	Escenario	V1	Respuesta del sistema	Resultado de la prueba
EC 2.1	EC 2.1 Representar satisfactoriamente edificio 3D.	V Edificio 106	El sistema muestra la representación visual de los objetos de tipo edificio aplicando un modelo de malla a partir datos almacenados.	Satisfactoria

4.8.2 Resultados de la aplicación de las Pruebas de Caja Negra.

La aplicación de las pruebas permitió comprobar el correcto funcionamiento del componente de visualización, las mismas estuvieron basadas en tres iteraciones, en la primera se detectaron dos no conformidades, una de interfaz de usuario y otra de funcionalidad, las cuales fueron corregidas. Posteriormente se realiza una segunda iteración en la que se detectaron dos no conformidades, una referente a un error ortográfico, y otra correspondiente a una funcionalidad, estas fueron solucionadas. Para finalizar se realiza una última iteración en la cual no se detectaron nuevas no conformidades.

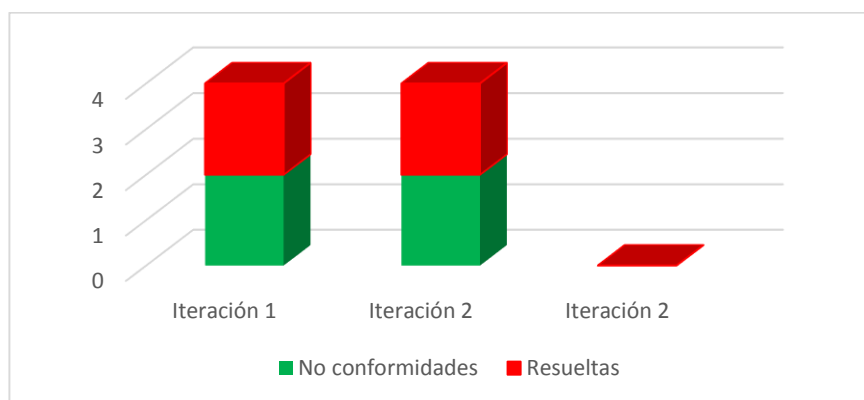


Fig. 8: Resultado de las pruebas de caja negra.

4.9 Conclusiones Parciales.

Sin lugar a dudas, la aplicación de la metodología de desarrollo de software permitió guiar el desarrollo del mismo. La aplicación de los estilos y patrones arquitectónicos así como el empleo de los patrones de

diseño permitió estructurar el software, al generar un diseño con un nivel de organización relevante, lo cual facilitó la construcción del mismo. Se realizaron pruebas de Caja Negra, empleando la técnica de partición equivalente la cual evidenció un grupo de 4 no conformidades las cuales fueron resueltas, este proceso permitió identificar los errores existentes en el componente para darle solución, se logró de esta forma que el producto cumpla las expectativas deseadas.

Conclusiones Generales.

1. El análisis de las formas de representar objetos en 3D en los SIG permitió apreciar las características de este tipo de sistemas para analizar su aplicabilidad en la solución del problema identificado.
2. Con la investigación se ratifica la importancia de los SIG y la posibilidad de usar la tecnología WebGL para la construcción de objetos 3D en entornos web, así como el empleo de librerías JavaScript que permitan agilizar este proceso, lo cual garantiza obtener soluciones que eliminen cualquier tipo de dependencias adicionales.
3. La investigación brinda los fundamentos necesarios sobre la representación en 3D de los edificios en el SIG UCI y refleja el cumplimiento de los requisitos funcionales.
4. La implementación del componente de visualización de objetos geo-referenciados en 3D crea las bases para el desarrollo de nuevas investigaciones vinculadas a la incorporación de la visualización tridimensional sobre los SIG.

Recomendaciones.

1. Dar inicio a otras investigaciones que permitan incorporar nuevas funcionalidades a la herramienta propuesta en función de mejorar la representación de las edificaciones geo-referenciadas en el SIG UCI a través de la visualización de estructuras geográficas en 3D.
2. Continuar profundizando en el tema y analizar nuevas tendencias y tecnologías que surjan a nivel mundial para identificar sus potencialidades y su posible aplicación.

Referencias

- ALARCÓN, G. T. Sistema de información geográfico turístico de Copacabana para la agencia Servimarter Tours, 2010.
- ALIAGA, G. Juan Peña Llopis. Sistemas de Información Geográfica aplicados a la gestión del territorio Revista de Geografía Norte Grande, 2006, (36): 97-101.
- ASHWORTH, S. Ext JS 4 Web Application Development Cookbook. Packt Publishing Ltd, 2012. p. 1849516863
- BASS, L. Software architecture in practice. Pearson Education India, 2007. p. 8177589962
- BUSCHMANN, F.; R. MEUNIER, et al. Pattern-oriented software architecture: a system of patterns. 1996 Part II, 2001.
- CANTOR, D. and B. JONES. WebGL beginner's guide. Packt Publishing Ltd, 2012. p. 1849691738
- CASAMOR, A. S. Los protocolos en las redes de ordenadores. Edicions UPC, 2003. p. 8483017482
- COMA, I.; R. RODRÍGUEZ, et al. ARTGRAPH: un entorno integrado de desarrollo y ejecución de aplicaciones 3D tiempo real Actas CEIG, 1998, 98.
- CONSORTIUM, O. G. OGC Standards, Revisado Marzo,(2014), desde< <http://www.opengeospatial.org/standards>, 2013.
- CRUZ, M. D. Metodología para dirigir la lectura de mapas Revista Educación y Sociedad, 2009, (3).
- DAVILA, Z. Z. Integracion de BPMN2. 0 en el análisis de comunicaciones, soporte sobre la herramienta case ORYX, 2013.
- DE LA TORRE, C.; U. ZORRILLA, et al. Y CALVARO J. 2010 Guía de Arquitectura N-Capas orientada al dominio con .NET 4.0 España: Microsoft Ibérica SRL, 2010.
- GAMMA, E.; R. HELM, et al. Design patterns: elements of reusable object-oriented software. Pearson Education, 1994. p. 0321700694
- GÁRATE, J. L. A.; M. C. PALOMINO, et al. Sistemas de Información Geográfica en la Gestión Integral del Litoral, Madrid, España: sn, 2009.
- GINESTÀ, M. G. and O. P. MORA. Bases de datos en PostgreSQL, Recuperado el, 2012.
- HARVEY, F. A primer of GIS: fundamental geographic and cartographic concepts. Guilford Press, 2010. p. 1609181085

- JACOBSON, I.; G. BOOCH, et al. El proceso unificado de desarrollo de software. Addison Wesley Reading, 2000. p.
- KEW, N. Desarrollo de módulos y aplicaciones con Apache. 2008. p. 8441523282
- KLIMKE, J. and J. DÖLLNER Service-Oriented Visualization of Virtual 3D City Models, 2011.
- KOLBE, T. H. Representing and exchanging 3D city models with CityGML. en: 3D geo-information sciences. Springer, 2009. 15-31.p.
- KRUCHTEN, P. The rational unified process: an introduction. Addison-Wesley Professional, 2004. p. 0321197704
- LARMAN, C. UML y Patrones. Pearson, 1999. p. 8420534382
- NAVARRO, A. P. Introducción a los sistemas de información geográfica y geotelemática. Editorial UOC, 2011. p. 8497889339
- NEUTENS, T. and P. DE MAEYER. Developments in 3D geo-information sciences. Springer, 2010. p. 3642047912
- OBE, R. and L. HSU. PostGIS in action. Manning Publications Co., 2011. p. 1935182269
- OLAYA, V. Sistemas de Información Geográfica Cuadernos Internacionales de Tecnología para el Desarrollo Humano, 2009, núm. 8, 2011.
- PATTERSON, T. C. Google Earth as a (not just) geography education tool Journal of Geography, 2007, 106(4): 145-152.
- PATTON, M. Q. Qualitative research. Wiley Online Library, 2005. p. 0470013192
- PÉREZ, J. E. Introducción a JavaScript Librosweb. es (editorial digital), 2009.
- PHP, M. P. S. Manual de PHP, 2006.
- PRESSMAN, R. S. Ingeniería del Software: Un enfoque práctico. 2005. p. 8448111869
- RAIDA, J. and C. ZERBST Perspectiva en 3D: escenas 3D en navegadores web con Three. js Linux magazine, 2013, (90): 33-38.
- REBOLLO, A. P. and M. C. FERNÁNDEZ Gestión Municipal a través de la Integración de los Estándares OGC en los Interfaces 3D, 2011.
- ROBINSON, A. H.; J. M. R. I VIVES, et al. Elementos de cartografía. Omega, 1987. p. 8428207682

Referencias

ROMEU, A.; D. GASTÓN, et al. Clientes IDE 3D: Visualización de edificios 3D y SIG móvil, 2012.

SELLERS, G.; J. OBERT, et al. Rendering massive virtual worlds. ACM SIGGRAPH 2013 Courses, ACM, 2013. 23 p. 1450323391

SHAW, M. and D. GARLAN. Software architecture: perspectives on an emerging discipline. Prentice Hall Englewood Cliffs, 1996. p.

SIEBER, R.; R. SCHNÜRER, et al. The Power of 3D Real-Time Visualization in Atlases—Concepts, Techniques and Implementation, 2012.

SOMMERVILLE, I. Ingeniería del software. Pearson Educación, 2005. p. 8478290745

TECHNOLOGIES, K. WebGL Earth 2013. [2013]. Disponible en: <http://www.webglearth.org/>

VIGNAGA, A. and D. PEROVICH Enfoque metodológico para el desarrollo basado en componentes Artículo Extraído el, 2003, 21.

WATSON, B.; P. MÜLLER, et al. Procedural Urban Modeling in Practice IEEE Computer Graphics and Applications, 2008, 28(3).

Bibliografía

- ALARCÓN, G. T. Sistema de información geográfico turístico de Copacabana para la agencia Servimarter Tours, 2010.
- ALIAGA, G. Juan Peña Llopis. Sistemas de Información Geográfica aplicados a la gestión del territorio *Revista de Geografía Norte Grande*, 2006, (36): 97-101.
- ASHWORTH, S. *Ext JS 4 Web Application Development Cookbook*. Packt Publishing Ltd, 2012. p. 1849516863
- BASS, L. *Software architecture in practice*. Pearson Education India, 2007. p. 8177589962
- BUSCHMANN, F.; R. MEUNIER, *et al.* Pattern-oriented software architecture: a system of patterns. 1996 *Part II*, 2001.
- CANTOR, D. and B. JONES. *WebGL beginner's guide*. Packt Publishing Ltd, 2012. p. 1849691738
- CASAMOR, A. S. *Los protocolos en las redes de ordenadores*. Edicions UPC, 2003. p. 8483017482
- COMA, I.; R. RODRÍGUEZ, *et al.* ARTGRAPH: un entorno integrado de desarrollo y ejecución de aplicaciones 3D tiempo real *Actas CEIG*, 1998, 98.
- CONSORTIUM, O. G. *OGC Standards*, Revisado Marzo,(2014), desde < <http://www.opengeospatial.org/standards>, 2013.
- CRUZ, M. D. Metodología para dirigir la lectura de mapas *Revista Educación y Sociedad*, 2009, (3).
- DÁVILA, Z. Z. Integración de BPMN2. 0 en el análisis de comunicaciones, soporte sobre la herramienta case ORYX, 2013.
- DE LA TORRE, C.; U. ZORRILLA, *et al.* Y CALVARO J. 2010 Guía de Arquitectura N-Capas orientada al dominio con .NET 4.0 *España: Microsoft Ibérica SRL*, 2010.
- GAMMA, E.; R. HELM, *et al.* *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994. p. 0321700694

GÁRATE, J. L. A.; M. C. PALOMINO, *et al.* *Sistemas de Información Geográfica en la Gestión Integral del Litoral*, Madrid, España: sn, 2009.

GARCIA, H. G. *Sistemas de Información Geográfica: Ordenamiento Territorial*, 2011. [2013]. Disponible en: <https://sites.google.com/site/sigarcgis/4-modelo-vector--raster/ventajas-y-desventajas>

GINESTÀ, M. G. and O. P. MORA. *Bases de datos en PostgreSQL*, Recuperado el, 2012.

HARVEY, F. *A primer of GIS: fundamental geographic and cartographic concepts*. Guilford Press, 2010. p. 1609181085

JACOBSON, I.; G. BOOCH, *et al.* *El proceso unificado de desarrollo de software*. Addison Wesley Reading, 2000. p.

KEW, N. *Desarrollo de módulos y aplicaciones con Apache*. 2008. p. 8441523282

KLIMKE, J. and J. DÖLLNER *Service-Oriented Visualization of Virtual 3D City Models*, 2011.

KOLBE, T. H. Representing and exchanging 3D city models with CityGML. en: *3D geo-information sciences*. Springer, 2009. 15-31.p.

KRUCHTEN, P. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004. p. 0321197704

LARMAN, C. *UML y Patronos*. Pearson, 1999. p. 8420534382

NAVARRO, A. P. *Introducción a los sistemas de información geográfica y geotelemática*. Editorial UOC, 2011. p. 8497889339

NEUTENS, T. and P. DE MAEYER. *Developments in 3D geo-information sciences*. Springer, 2010. p. 3642047912

OBE, R. and L. HSU. *PostGIS in action*. Manning Publications Co., 2011. p. 1935182269

OLAYA, V. *Sistemas de Información Geográfica Cuadernos Internacionales de Tecnología para el Desarrollo Humano*, 2009, núm. 8, 2011.

- PATTERSON, T. C. Google Earth as a (not just) geography education tool *Journal of Geography*, 2007, 106(4): 145-152.
- PATTON, M. Q. *Qualitative research*. Wiley Online Library, 2005. p. 0470013192
- PÉREZ, J. E. Introducción a JavaScript *Librosweb. es (editorial digital)*, 2009.
- PHP, M. P. S. *Manual de PHP*, 2006.
- PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. 2005. p. 8448111869
- RAIDA, J. and C. ZERBST Perspectiva en 3D: escenas 3D en navegadores web con Three. js *Linux magazine*, 2013, (90): 33-38.
- REBOLLO, A. P. and M. C. FERNÁNDEZ Gestión Municipal a través de la Integración de los Estándares OGC en los Interfaces 3D, 2011.
- ROBINSON, A. H.; J. M. R. I VIVES, *et al. Elementos de cartografía*. Omega, 1987. p. 8428207682
- ROMEU, A.; D. GASTÓN, *et al. Clientes IDE 3D: Visualización de edificios 3D y SIG móvil*, 2012.
- SELLERS, G.; J. OBERT, *et al. Rendering massive virtual worlds*. ACM SIGGRAPH 2013 Courses, ACM, 2013. 23 p. 1450323391
- SHAW, M. and D. GARLAN. *Software architecture: perspectives on an emerging discipline*. Prentice Hall Englewood Cliffs, 1996. p.
- SIEBER, R.; R. SCHNÜRER, *et al. The Power of 3D Real-Time Visualization in Atlases—Concepts, Techniques and Implementation*, 2012.
- SOMMERVILLE, I. *Ingeniería del software*. Pearson Educación, 2005. p. 8478290745
- TECHNOLOGIES, K. *WebGL Earth* 2013. [2013]. Disponible en: <http://www.webglearth.org/>
- VIGNAGA, A. and D. PEROVICH Enfoque metodológico para el desarrollo basado en componentes *Artículo Extraído el*, 2003, 21.

Bibliografía

WATSON, B.; P. MÜLLER, *et al.* Procedural Urban Modeling in Practice *IEEE Computer Graphics and Applications*, 2008, 28(3).